MODELING HIGH DIMENSIONAL MULTI-STREAM DATA FOR MONITORING AND PREDICTION

A Dissertation
Presented to
The Academic Faculty

By

Samaneh Ebrahimi

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
H. Milton Stewart School of Industrial & Systems Engineering (ISyE)

Georgia Institute of Technology

December 2018

MODELING HIGH DIMENSIONAL MULTI-STREAM DATA FOR MONITORING AND PREDICTION

Approved by:

Dr. Kamran Paynabar
H. Milton Stewart School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Jianjun Shi
H. Milton Stewart School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Nagi Gebraeel
H. Milton Stewart School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Chuck Zhang
H. Milton Stewart School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Shawn Mankad
Samuel Curtis Johnson Graduate School of Management
Cornell University

Date Approved: August 14, 2018

To my beloved parents, Sorayya and Firouz,

my twin sister Marzieh,

and my siblings Nafiseh, Amir, and Ebi

for their love and endless support

ACKNOWLEDGEMENTS

My sincere gratitude goes to my advisor, Professor Kamran Paynabar, for his splendid supervision. His openness to hearing and encouraging new ideas helped me explore new fields and gain self-confidence in doing this research. Besides, his continuous support during my graduate life made my Ph.D. journey much easier.

I want to give my special thanks to all other committee members, Professor Jan Shi, Professor Nagi Gebraeel, Professor Shawn Mankad, and Professor Chuck Zhang, for their invaluable comments, suggestions, and guidance during this dissertation. Moreover, my special gratitude goes to my former undergraduate advisor, Professor Hashem Mahlooji. His strong confidence in me is one of the reasons that I pursued graduate studies and, for which I will always be grateful.

Besides, I'd like to thank my dear friends who were indeed my second family during my life and helped me go through this journey. To name a few, I would like to thank Dr. Shaghayegh Fathi, Neda Mohsenianrad, Golnaz Tehranchi, Dr. Chitta Ranjan, Sahba Akhavan Niaki, Dr. Negin Enayaty Ahangar, Dr. Satya Malladi , Geet Lahoti, Dr. Amirebrahim Darabi, Mostafa Reisi Gahrooei, Yasaman Mohammad Shahi, Ali Namayandeh, Shaghayegh Navabpour, Arezoo Shirazi, and Shiva Bahrami.

Finally, I want to express my deep gratefulness to my precious parents for their endless love and support which gave me the will and strength to keep up in all of my life. Also, delightful thanks to my beloved twin sister Marzieh, my dear siblings Nafiseh, Amirhossein, Ebrahim, and my new siblings Sepideh, Arash, and Vinod.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This dissertation concentrates on solving problems for monitoring and predictions of high dimensional, streaming data using new data mining methods. Chapter 1 illuminates the need for new approaches for managing such data. Of the plethora of problems that exist, , this dissertation attempts to focus on three distinct and critical research problems. In Chapter 1, we concisely review the motivation and challenges behind each problem.

In Chapter 2, we propose a new monitoring and diagnosis approach based on PCA for monitoring high-dimensional, multi-stream data. Monitoring and diagnostics (M&D) are important components of Statistical Process Control (SPC). However, little work exists for an integrated M&D approach. M&D's main challenge is handling high-dimensional processes commonly found in manufacturing, computer networks, and the Internet industry. In Chapter 2, we propose an integrated approach that addresses this challenge. For monitoring, the most commonly used methods in high dimensions are based on finding the underlying lower dimension. One common approach is Principal Component Analysis (PCA). For PCA-based monitoring, selecting the Principal Scores (PCs) to include in the model is important. While most of the existing methods focus on PCs with the highest variance, we argue that this is an inappropriate approach for the purpose of monitoring. Quite the opposite, we show that adaptively chosen PCs are significantly better for process monitoring. Consequently, we develop a novel monitoring method based on this principle named Adaptive PC Selection (APCS). More importantly, we integrate a novel diagnostic approach to enable a streamlined SPC. The PC-based Signal Recovery (PCSR) diagnostics approach draws inspiration from Compressed Sensing to use Adaptive Lasso for identifying the sparse change in the process. We theoretically motivate our approaches and do performance evaluation of our integrated M&D method through simulations studies under various scenarios and two case studies.

In Chapter 3, we propose a new methodology for dynamically monitoring sparse at-

tributed networks. For this, we mainly focus on modeling the network connections in financial institutions. The interconnectedness of financial institutions can function as a mechanism for the propagation and amplification of shocks throughout the economy, thus contributing to financial crises. As such, network analysis has become a critical tool for assessing interconnectedness and systemic risk levels. In Chapter 3, we create a formal monitoring system to detect changes within a sequence of sparse networks constructed from an interbank lending market in the European Union. The approach combines a state space model with the Hurdle model to capture temporal dynamics of the edge formation process, which is modeled as a function of node and edge attributes and estimated using an extended Kalman Filter. Moreover, statistical process control charts such as Exponential Weighted Moving Average (EWMA) control charts, are used to monitor the network sequence in real time in order to distinguish the gradual change resulting from the typical edge dynamics from abrupt changes in trading patterns caused by fundamental changes in market conditions. We find that the proposed methodology would have raised alarms for regulators prior to several key events and announcements by the European Central Bank during the 2007-2009 financial crisis, demonstrating the promise of the approach as an early warning system. Moreover, we show the efficacy of our proposed approach using simulation studies over various scenarios.

In Chapter 4, we propose a novel deep learning approach for predicting multimedia data labels. The method is the extension to the Classification Restricted Boltzmann Machine (ClassRBM). The Restricted Boltzmann Machine (RBM) is a probabilistic model used to model the distribution of a visible layer of features using one hidden layer. Deep Boltzmann Machines were developed as an extension of RBM with multiple hidden layers. These methods have been successfully applied for unsupervised learning. A new discriminative RBM for supervised learning known as Classification RBM (ClassRBM) was proposed in 2008 [68]. Due to estimation intractability, an effective deep extension of ClassRBM has not been used in the literature. In this chapter, we propose a new estimation

approach for deep ClassRBM (ClassDBM). Moreover, inspired from multimodal DBMs [107], we show that ClassDBM can be extended to multimodal ClassDBM. Lastly, we implement the proposed approach on two benchmark data and advertisement multimedia data for validation.

Lastly, Chapter 5 concludes the dissertation by summarizing the research contributions, outcomes, and future directions.

CHAPTER 1
INTRODUCTION AND BACKGROUND

1.1 Background

In the era of Big Data, the amount of data produced is increasing exponentially [81, 41]. High-dimensional, streaming, and multimedia data are prevalent in a wide variety of fields, including manufacturing, healthcare, advertising, the financial and service industries, and social networks. One of the leading challenges in this era is developing scalable and effective methodologies for precise modeling and meaningful interpretation of such complex data. These models can facilitate gaining insights about data and reaching important decisions. For example, in manufacturing and network systems, efficient monitoring and fast detection of out-of-control situations are crucial for taking proper corrective actions to avoid failures and catastrophic events. In the service industries, analysis of streaming data helps to gain a proper interpretation of system and customer behaviors, resulting in the ability to provide better quality service. Nonetheless, analysis of contemporary data is challenging due to its high dimensionality, lack of structure, and auto- and cross-correlation, as well as the evolving nature of data itself. The primary goal of this dissertation is to propose new statistical learning methodologies for scalable modeling of high-dimensional data for monitoring and prediction that overcome these challenges. Hence, this dissertation concentrates on three particular problems and presents statistical models for solving each problem. In Section 1.2, different data types studied in this dissertation and their challenges are explained. Afterwards, in Section 1.3, the research topics are explained, and the motivation behind each problem, the challenges, and the proposed solutions are briefly elucidated.

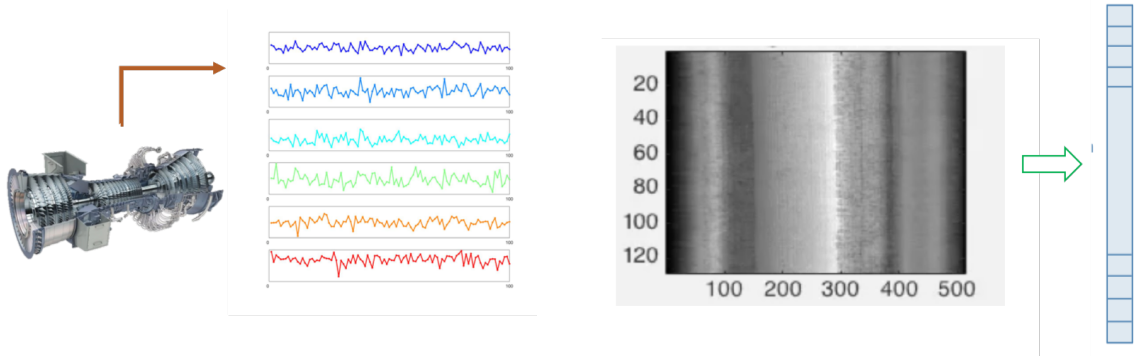
1.2 Data Types and Challenges

1.2.1 High Dimensional, Multistream Data

With the rapid advancements in machinery, data-sensing systems are now capable of recording real-time data with multiple attributes. Hence, every second, a large amount of streaming data (i.e. manufacturing quality data, network flows, and multimedia streams) is being generated. Mining and exploring this streaming data is challenging due to its high dimensionality, evolving nature, and auto- and cross-correlation. These challenges have motivated researchers to develop new methodologies that can model the data over time and provide useful interpretations. Many recent papers focus on mining such real-time streaming data [96, 2]. Two critical challenges about this data are a) high dimensionality and b) complex correlation structure. High dimensionality, combined with the substantial number of observations, causes traditional methods to fail in modeling the data. Hence, we require statistical models that are scalable—i.e., being able to maintain high efficiency and precision as the number of attributes and observations grows. Moreover, as the dimension grows, the correlation structure gets more complicated. Besides, since data is collected over time, each observation has a temporal correlation with the past observations. For example, in the steel rolling process presented in Figure 1.1 (b), an image can be representative of the process at each time frame. In this case, image pixels are spatially correlated while they have a temporal correlation with the other images over time. The correlation structure raises the need for new models capable of capturing this structure correctly.

1.2.2 Network Stream Data

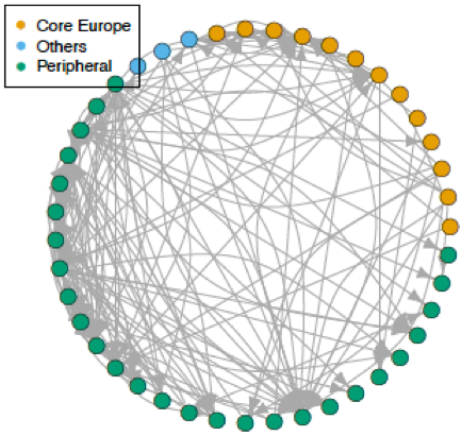
In recent years, networks and network studies have attracted considerable interest in various fields. Presenting a system as a network aids in representing the elements in the system and the connections and interactions among them in a meaningful way. The network representation is robustly expressive and can capture complex relationships. Hence, with the



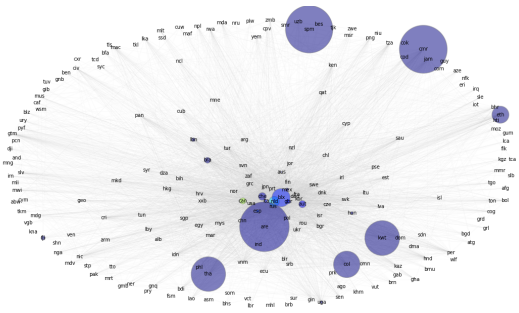
(a) Gas Turbines (b) Steel Rolling Process

Figure 1.1: Examples of High Dimensional Streaming Data

rapid advancements in data collection, network analysis emerges in diverse areas such as biology, economics, manufacturing, sociology, and more. The diversity of network examples is as great as the number of ways people or entities interact with each other. These examples include social networks like Facebook and Twitter, email, citation and authorship networks, protein interactions networks, process networks, and so on[43]. One major challenge in modeling network data arises from the fact that entities in the network are highly correlated and connected with each other. Moreover, the nature of relations among entities in the network can be complicated and not straightforward to capture and interpret. Therefore, an effective statistical model must be able to model networks such that all relations are captured in the model. Moreover, network data are mostly high dimensional and in considerable quantity, with a complicated correlated structure; thus, as discussed earlier, a scalable method that can capture the complex correlation structure is essential [47]. Another important characteristic of networks is their dynamic nature. Networks are constantly changing over time. These changes can occur gradually or suddenly. Traditional statistical methods are not able to capture the dynamic nature of a network properly. To address the challenges above, innovative analysis methods are required.



(a) European Interbank Network



(b) 2014 Global Import4 Commodity Network

Figure 1.2: Examples of networks and multimedia data

1.2.3 Multimedia: Image, Audio, Text, Etc.

Multimedia data consist of one or more media data types such as text, images, speech, audio, and video. With the advancements in data collection and storage, the amount of multimedia data available in different industries is growing rapidly. Multimedia data can be in the form of manufacturing data, advertising data, surveillance videos, meetings records, traffic records, news, medical data, etc. There is much more knowledge and information stored in multimedia data in comparison with simple high-dimensional vector data. Hence, with the ubiquitous presence of this data, Multimedia Data Mining (MDM) has gained popularity among researchers. MDM includes mining any media type separately or together at a time. Mining multimedia data can help discover the information and patterns in such data [12].

Mining multimedia data can be quite challenging because of its characteristics. These characteristics can be summarized as below:

- **Lack of structure**. Multimedia data are unstructured or semi-structured. Most traditional data mining approaches are designed for structured data. To be able to work with Multimedia data, we need to convert it into a structured format.

- **High dimensionality and complexity**. Multimedia data is extremely high dimen-

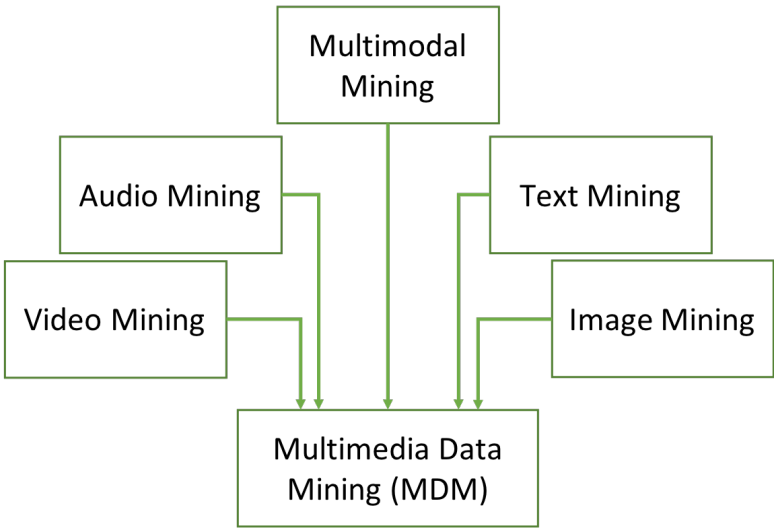


Figure 1.3: Multimedia Data Mining

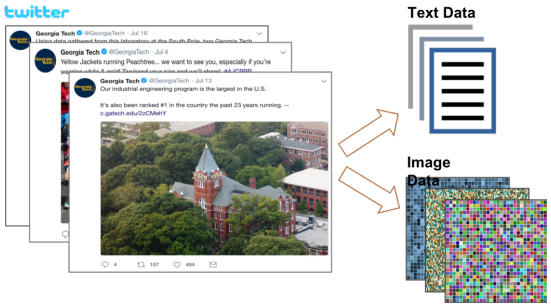
sional, and the number of observations is usually massive. This results in a need for feature extractions and dimension reduction techniques that are scalable. Moreover, the information stored in such data is very complicated. Typically, we need models with many parameters that can cause overfitting on such data.

- **Multiple data modals**. In the multimedia era, data is typically collected in multiple types. For example, for surveillance data, we can have images, audio files, and video. With each modality needing its own pre-processing and transformation methods, it is perplexing to find the connection and correlation among different modalities.

- **Interpretation**. Mining multimedia data can lead to complicated models that are not easy to interpret. There can be several different interpretations from a single model due to the complicated nature of multimedia data. To overcome this, we need a model that can help us gain a better interpretation.

To address the challenges mentioned above, this dissertation concentrates on three problems: Large-streaming data analytics for monitoring and diagnostics in manufacturing Systems; Dynamic network monitoring using extended kalman filter on Hurdle models; Discriminative DBM for classification and its extension to multimodal inputs.

(a) Online Advertisement Data



(b) Twitter Multimedia Data

Figure 1.4: Examples of Multimedia Data

1.3 Dissertation Research Topics

In this section, the research topics in each chapter are briefly explained.

1.3.1 Large-Streaming Data Analytics for Monitoring and Diagnostics in Manufacturing Systems

The problem of process monitoring and fault diagnosis using high-dimensional (HD) streaming data has attracted increasing attention. In an HD system, real-time modeling, monitoring, and prompt detection of changes in the system are crucial. More importantly, after detecting a change in the system, a diagnosis study of the faulty variables is necessary to prevent the system's breakdown and provide insights into its improvement. An example of high-dimensional streaming data can be found in the gas turbine systems used for power generation shown in Figure 1.1 (a). In such systems, the performance of a confined combustion process is being monitored using hundreds of sensors measuring temperature, vibration, pressure, etc. in different chambers and segments of the turbine. Utilizing the information in the sensors, monitoring and instant detection of any changes, followed by the diagnosis of the faulty variables, is necessary to inhibit imminent blowout that leads to lengthy shutdown and repair and therefore can result in a tremendous cost.

Although Monitoring and Diagnostics (MD) of high-dimensional systems is crucial,

little work exists for developing an integrated MD approach that takes into account the high dimensionality and correlation among data. In this chapter, we propose an integrated approach that addresses this challenge. For monitoring HD data, the most commonly used methods are based on extracting low-dimensional features. One popular approach is to implement Principal Component Analysis (PCA). PCA is a widely known projection method that transforms dependent data into uncorrelated features known as PC scores. Also, in PCA-based monitoring methods, a few PCs are selected as monitoring features. Hence, PCA can address the dependency and high-dimensionality issue.

In most of the existing methods, PCs with the highest variance are selected as monitoring features [57, 121, 94, 75, 73]. In this chapter, we argue that despite the popularity of this approach, considering the top PC scores with the highest variance as monitoring features may not always be the right approach for process monitoring because small changes may be masked by the variation present in them. Afterwards, we show that adaptively selecting PCs is significantly more effective for process monitoring. Therefore, an Adaptive PC selection (APC) approach is presented. The APC method based on hard-thresholding for selecting the set of PC scores most susceptible to an unknown change is presented. Unlike the top-PCs approach, in our approach, the number of features may vary at each sampling time and is adaptively selected based on the observed sample and its standardized distance from the in-control mean. The proposed approach does not require any prior knowledge about a fault or change direction, which makes it more universally applicable.

Another issue with PCA-based monitoring methods is the lack of diagnosability. This is because the PC scores used as monitoring features are linear combinations (additive statistics) of original measurements. Therefore, if a PC score initiates an out-of-control alarm, it would be difficult to attribute the score to any specific process variables. Interpretation and decomposition of these additive statistics are often theoretically difficult and/or computationally expensive in HD data streams. To encounter this problem, we propose a novel PC-based Signal Recovery (PCSR) diagnostics approach that seamlessly integrates with

the PCA-based monitoring methods. This approach draws inspiration from Compressed Sensing and uses Adaptive Lasso to identify the subset of altered variables. Moreover, we theoretically prove the consistency of our proposed diagnostics method. Finally, we evaluate the performance of our integrated MD method through simulations and two case studies.

1.3.2 Dynamic Network Monitoring Using Extended Kalman Filter On Hurdle Models

Representing systems as networks is a robust way to illustrate entities and their complicated relationships and connections. Since relationships among entities can evolve, an adequate network representation should capture the dynamic in the system. Nowadays, dynamic networks can be found in a wide variety of areas. Examples include financial relationships between banks, social networks, relationships among countries, etc. One critical problem in dynamic networks is the change point detection, i.e., detecting a time at which the network has deviated significantly from its normal condition. The dynamic nature of traditional network monitoring methods makes it difficult for them to separate dynamic and structural changes and to capture and detect changes efficiently [97]. An example of a dynamic network is the inter-bank lending market in the European Union. Monitoring such networks is essential to detecting any abnormal changes in the network instantly. Prompt change point detection can help in identifying the onset of a financial crisis. The 2007 financial crisis imposed a considerable amount of cost regarding economic growth and direct bailouts. Although the finance network analysis methods in the literature have established the importance of network statistics for early warning systems, they were not able to develop a methodology that systemically identifies whether the market has entered a new epoch in real time. This is an especially significant problem in practice given that high false positive rates that can occur unless a careful approach is utilized that can distinguish gradual change resulting from the typical edge dynamics from abrupt changes caused by economic shocks to the financial system.

In real-world network examples, there is massive sparsity in the network edges. In other words, among the nodes, there are many pairs of nodes that are not connected. Generalized Linear Models such as Poisson models cannot take into account these excessive zeros in communications. To solve this issue, we propose the use of Hurdle regression modeling. The Hurdle model is a two-part model for count data where the occurrence of zero counts is separated from the occurrence of positive counts. Hence, the first part in the model is a binary model, such as logistic or probit regression, for modeling if the count is zero or greater than zero. If the first part identifies a positive count, then the second part of the model which is a positive count distribution, such as positive Poisson regression, models the positive number of occurrence.

Furthermore, most financial networks are dynamic, meaning that their structure gradually changes over time. This gradual change should be modeled and distinguished from any sudden changes. To capture the temporal dynamics of the edge formation process, we integrate the Hurdle model with state-space models and use the Extended Kalman Filter to update the model parameters recursively over time.

Finally, we generate a one-step-ahead prediction of the network and compare it to the realized network to decide whether the observed evolution was smooth or abrupt using an Exponentially Weighted Moving Average (EWMA) control chart. The efficacy of our proposed method is shown with simulation studies and with an application on the European finance data.

1.3.3 Discriminative DBM For Classification And Its Extension To Multimodal Inputs

When working with multimedia data, general Machine Learning (ML) approaches are not capable of extracting meaningful features and modeling the complicated nature of this type of data. Hence, more complex models are required. One example of multimedia data is advertising data from online music streaming companies. In such, the data consists of multiple modalities such as images, audio files, and scripts of audios as shown in Figure 1.4

(a). Ensuring that ads have the proper quality establishes a better user experience and hence results in longer user engagement. Therefore, designing a model that can automatically assess the ads and predict their quality can help in better creation of ads as well as better ranking of available ads.

Hence, measuring and predicting the effectiveness of an ad is crucial. However, developing a meaningful model over this data and gaining insights on how to create an effective ad are not straightforward. Some challenges in modeling advertising data are the unstructured essence of multimedia data; the data's high dimensionality and complicated correlation structure; and the presence of different modalities.

One approach to modeling multimedia data is the use of a Restricted Boltzmann Machine (RBM). RBM is a probabilistic model with stochastic visible units connected to stochastic binary hidden units [37]. The hidden units can be used as a simpler representation of the complex visible inputs. Therefore, RBM is a generative model used to feature engineering and dimension reduction [54]. RBMs are applied extensively to different types of data such as images [111, 66], speech [58, 76], text documents [24], etc. However, similar to many existing Machine Learning methods, RBMs have a "shallow" architecture, meaning they usually have only one layer of hidden units. Systems with "shallow" architectures are not able to extract complex structures from the input, especially when the input has an intricate nature [10]. Therefore, multilayer generative models were introduced to better represent complicated data. So, RBMs were extended to deep networks such as Deep Boltzmann Machine (DBM) [98]. The studies show that these deep networks outperform traditional ML approaches like Support Vector Machines (SVM) and traditional feedforward neural networks in modeling high-dimensional complex data [62] .

Although RBMs and DBMs are generally used as feature extractors for complex data, or as initializers for other classification deep neural network methods, these methods cannot be directly applied to a classification (discriminative) problem. To extend RBM methods to discriminative problems, the classification RBM (ClassRBM) was proposed [68]. The

ClassRBM was used extensively in the literature for various classification problems, such as author profiling using emails and blogs [5], radar target recognition [92], breast cancer prediction, classification problems in medical domains [113, 112], credit risk classification [74], and so on. In spite of the frequent implementation of ClassRBM in different fields, the extension of this method to a deeper architecture is not available in the literature. The main reason is that the learning model in the ClassRBM approach will become intractable when more than one hidden layer is introduced. Hence, to the best of our knowledge, extending the method to deeper networks is not available in the literature.

In this chapter, we aim at extending ClassRBMs to a deeper architecture. Specifically, we propose a novel estimation approach for deep ClassRBM (classDBM) when we have more than one hidden layer. In the proposed method, we present novel algorithms for learning the intractable problem and for predicting new observations after learning the model. Moreover, we extend the proposed method for multimodal inputs. For comparison study, we implement our proposed method on two benchmark datasets (MNIST, and NORB object recognition) used extensively in the literature. We evaluate the prediction accuracy of our ClassRBM by comparing it to traditional deep learning methods. Besides, we implement the proposed method on audio advertising data and present insights on designing an effective audio ad.

CHAPTER 2
LARGE-STREAMING DATA ANALYTICS FOR MONITORING AND DIAGNOSTICS IN MANUFACTURING SYSTEMS

2.1 Introduction

Recently, the problem of process monitoring and diagnosis using a large set of multi-stream data has become a research focus in the field of statistical process control (SPC). The reason is two-fold; first, in the past decade, sensing technologies have enabled fast measurement of a large number of process variables, resulting in large data streams, and, second, Conventional multivariate methods such as Hotelling's T^2, MEWMA, and MCUSUM ([120, 106]) are not appropriate for analyzing large streaming data due to their lack of scalability in terms of both the computational time and the detection power. An example of large streams can be found in gas turbine systems used for power generation. In these systems, the performance of the confined combustion process is being monitored using hundreds of sensors measuring temperature, vibration, pressure, etc. in different chambers and segments of the turbine. Early detection of any changes in the system, followed by the diagnosis of the faulty variables is necessary to avoid imminent blowout that leads to relighting the combustor and costly shutdowns. Another application of large streaming data is in image-based process monitoring in which each pixel of an image can be considered as a single data stream. For example, in a rolling process where the rollers are employed to lesson the cross-section of a long steel bar by applying pressing forces, the quality of produced bars can be inspected by a vision system that is set up to take images of the bar surface at short time intervals. A sample of such an image is shown in Figure 2.1. In this image, each row contains 512 pixels, and each pixel can be considered as a variable resulting in an HD correlated data stream.

Despite its importance, existing SPC literature lacks a scalable integrated M&D approach using large data streams. Conventional multivariate control charts can only monitor small or moderate data streams, effectively, and their performance quickly deteriorates as the number of data streams increases. To address the high-dimensionality issue, more recent works have focused on employing variable selection techniques to reduce the dimensionality by removing the set variables that are less susceptible to the process change. Examples of the variable-selection-based method include [117, 129, 19]. However, these methods are not scalable and generally require much more computational effort as the dimension grows. Moreover, most of these methods are not easy to be comprehended by an operator or process engineer.

There is a group of salable multivariate monitoring methods that are developed based on the assumption that data streams are independent and that the set of altered variables is sparse, meaning that only a small subset of variables is affected by a process change. For example, assuming that exactly one variable changes at a time, [110] proposed an approach based on the maximum of CUSUM statistics from each individual data stream. Mei [85, 86] developed a robust monitoring procedure based on the sum of (the top-r) local CUSUM statistics assuming all variables are independent and measurable. For the case that variables are not easily or efficiently measurable, [77] presented TRAS (top-r based adaptive sampling), which is an adaptive sampling strategy that uses the sum of top r local CUSUM statistics for monitoring. The sparsity assumption is generally valid in practice as a change or fault often affects only a small subset of variables. However, although theoretically and computationally appealing, the independence assumption is typically unrealistic.

To address the dependency and high-dimensionality issues, Principal Component Analysis (PCA) has been broadly used for monitoring multivariate data streams. PCA is a common projection method that transforms dependent data to uncorrelated features known as Principal Component (PC) scores. Often, only a few PC scores that explain the most variation of original data are used as monitoring features resulting in dimension reduction [57,

121, 94, 75, 73]. However, considering top PC scores with the highest variance as monitoring features may not always be a right approach for process monitoring. As an example, consider a bivariate normal distribution given in Figure 2.2, in which PC1 represents the direction of the eigenvector with larger eigenvalue, and the red arrow indicates the direction of change in the mean of the distribution. As can be seen from the figure, the effect of the change on both PC-scores is the same. However, the fact that PC1 constitutes the most of the process variance makes it less sensitive to the change compared with PC2. In other words, small changes may be masked by the variation present in top PC scores, hence becoming undetectable. Other PC selection criteria for process monitoring include the variance of reconstruction error (VRE) approach by [27], and the fault signal to noise ratio (SNR) by [125] and [109]. The VRE method selects a subset of PCs which minimizes fault reconstruction error, while the fault SNR method is based on fault detection sensitivity. The main disadvantage of these methods is that they require prior knowledge of the fault direction. In this paper, we present an adaptive PC Selection (APC) approach based on hard-thresholding for selecting the set of PC scores that are most susceptible to an unknown change. Unlike, the top-r-PCs approach, in our approach the number of features may vary at each sampling time and is adaptively selected based on the observed sample and its standardized distance from the in-control mean. The proposed APC approach does not require any prior knowledge about a fault or change direction, which makes it more universally applicable.

Another long-standing issue with PCA-based monitoring methods is the lack of diagnosability. This is because that the PC scores used as monitoring features are linear combinations (additive statistics) of original measurements. Therefore, if a PC score initiates an out-of-control alarm, it would be difficult to attribute it to any specific process variables. Interpretation and decomposition of these additive statistics are often theoretically difficult and/or computationally expensive in HD data streams. For diagnostics on a PC-based monitoring, one common approach is the use of contribution plots that specifies the contribution

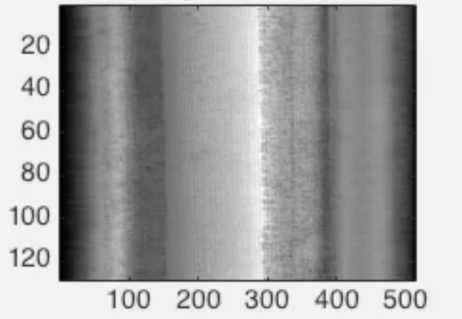


Figure 2.1: Surface image of steel bar in the rolling process

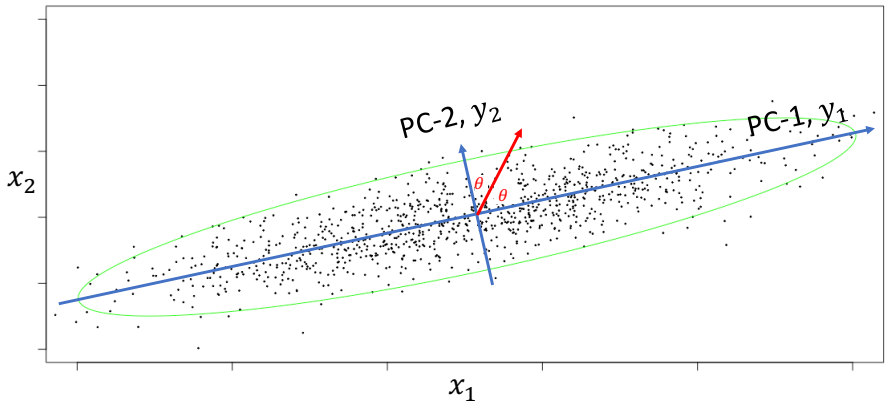


Figure 2.2: Example of a change having the same angle with both PCs

of each variable corresponding to the out of control statistic [119, 4, 60, 95]. Contribution plots are popular because of their ease of implementation and their ability to work without any a priori knowledge. However, where there are multiple faulty variables, the precise fault isolation is not promised using contribution plots [126]. To overcome this problem, hierarchical contribution plots was proposed [80]. However, it will perform poorly if the initial partitioning is not correct. Moreover, in the context of HD data, these methods will become difficult to interpret and are also computationally expensive.

For the purpose of diagnosis in multivariate control charts with original measurements, [117] and [129] proposed variable selection techniques. Both methods optimize a penalized likelihood function for multivariate normal observations to identify the subset of altered variables. The $\mathcal{L}1$-penalized regression method of [129] provides more compu-

tational advantages in implementation. [128] combined Bayesian Information Criterion (BIC) with penalization techniques to assist the fault localization process and suggested an Adaptive Lasso-based diagnostic procedure. However, these methods assume that the change point is already known and focus only on diagnosis. Additionally, they cannot easily be integrated with a PCA-based monitoring approach. To address these shortcomings, we propose a new diagnostics approach that seamlessly integrates with our proposed PCA-based monitoring method. The developed approach draws inspiration from Compressed Sensing and uses Adaptive Lasso to identify the subset of altered variables. For this purpose, we focus on detecting mean shifts, and we assume that change is sparse, i.e., only a small set of variables contribute to the change. As mentioned earlier, this is a reasonable assumption because in real-world, usually a small number of variables change.

The major contributions in this paper are, a) countering the traditional view of top-PC scores as the best method for process monitoring, and proposing an adaptive PC selection approach as an alternative; and b) proposing a new diagnostics approach that integrates with the proposed PCA-based monitoring framework An overview of the proposed Monitoring and Diagnostics (M&D) approach is shown in Figure 2.3 .

The rest of the paper is as organized as follows. Next section provides an overview of the proposed integrated monitoring-diagnostics framework. Then, the novel APC selection approach is presented and is integrated with an EWMA control chart. After that our proposed diagnostics approach based on adaptive lasso on PC scores is elaborated. The next section will present simulation studies for different scenarios, and performance analysis of our method in comparison with a few existing methods as benchmarks. Then, in the consequent section, using two case studies, we show that the proposed methods significantly outperform the benchmarks in terms of quick change detection (monitoring) and identification of the changed variables (diagnostics). Finally, we conclude and provide future research directions.

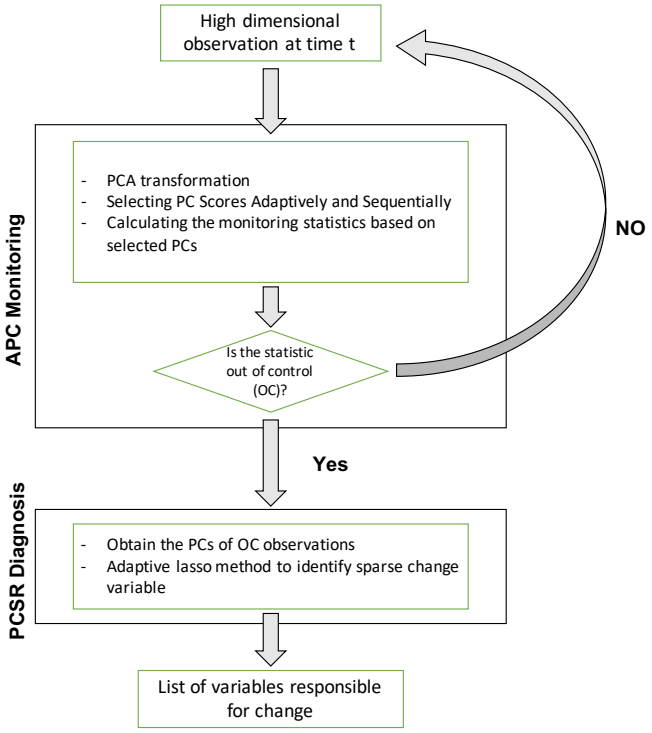


Figure 2.3: Methodology overview

2.2 Integrated PCA-Based Monitoring and Diagnostics for Large Data Streams

2.2.1 Background

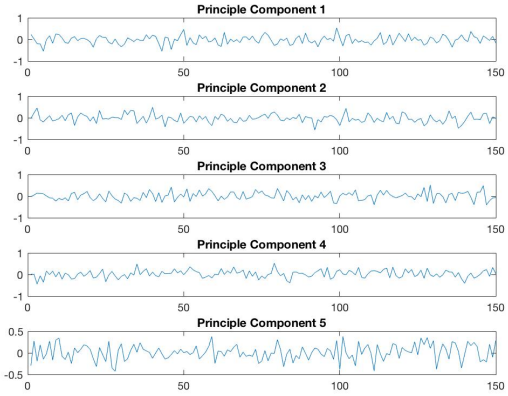
PCA is a linear transformation widely used for dimension reduction and generating uncorrelated features. Suppose for monitoring a process, a p-dimensional data stream denoted as $X = \{\mathbf{x}^{(i)} : \mathbf{x}^{(i)} \in \mathbb{R}^p;\ i = 1, 2, \ldots\}$ is collected at sampling time i. Without loss of generality, assume the data streams are centered (zero mean) with the covariance matrix $\mathbf{\Sigma}$. By applying PCA, this set of correlated observations can be converted into a set of linearly uncorrelated variables known as *principal component scores*, which can be computed by $\mathbf{y} = A^T\mathbf{x}$, where $A \in \mathbb{R}^{p\times p}$ is the matrix of eigenvectors of $\mathbf{\Sigma}$ and $\mathbf{y} \in \mathbb{R}^p$ are PCs. Also, it can be shown that $var(\mathbf{y}_j) = \lambda_j$ and $cov(\mathbf{y}_j, \mathbf{y}_k) = 0,\ \forall (j,k) \in \{1, \ldots, p\},\ j \neq k$. For ease of interpretation, the eigenvectors in A are arranged such that their corresponding eigenvalues are in decreasing order, i.e.m $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$. This ordering will be further referred throughout the paper for developing our methodology. In this paper, we call

the principal scores corresponding to higher and lower eigenvalues as top PCs and bottom PCs, respectively.

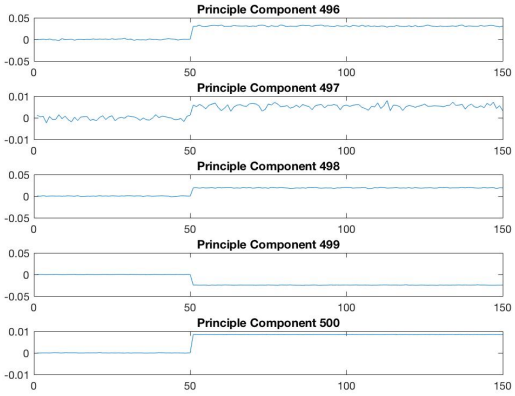
In most conventional PCA-based methods, top k PCs are selected for monitoring because they contain more process information. This approach, however, may not always result in proper monitoring features. To illustrate this, we synthesized 50 in-control samples of correlated data from a multivariate normal distribution with $\boldsymbol{\mu} = \mathbf{0}$, $\sigma = 0.1$, and the size of $p = 500$, followed by 100 out-of-control samples. In the vector of the out-of-control mean, 10% of elements have been randomly selected and shifted to 0.05σ. Using the covariance matrix used for generating data, we perform PCA and monitor all PCs separately. Figure 2.4 shows the control charts for top 5 and bottom 5 of PCs. As shown in the figure (left), the top PCs fail to detect the change. As mentioned earlier, this is due to the fact that top PCs have large variances, which make them insensitive to small shifts in the mean. On the other hand, the bottom PCs with smaller variances are more sensitive and can detect the small shift at the time it occurs, i.e., $t = 50$. This experiment was repeated several times, and each time similar results were. This shows that depending on the direction and the size of a change, the traditional approach of selecting top PCs may severely underperform. Therefore, it is imperative to propose a PC selection approach that can adaptively select the set of most sensitive PCs and does not depend on the a priori knowledge about the direction of the change.

2.2.2 Adaptive PC selection (APC) for Process Monitoring

In this section, we propose an adaptive PC Selection approach for process monitoring. This approach simply selects and monitors a set of PCs that shows a large deviation from a known in-control state. Suppose, the in-control observations follow $\mathbf{x}_i \sim N(\mathbf{0}, \boldsymbol{\Sigma}); i < \tau$, and at an unknown time τ a mean shift occurs such that $\mathbf{x}_i \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}); i \geqslant \tau$, where, $\boldsymbol{\mu}$ is a non-zero sparse vector, and the process covariance is assumed to remain constant. Therefore, given the eigenvector matrix $A \in \mathbb{R}^{p \times p}$, the PC scores after the process change



(a) top 5 PCs



(b) low 5 PCs

Figure 2.4: Comparing the behavior of top and low PCs for monitoring when a sparse shift happens in a random set of process of variables

will become $\mathbf{y_i} = A^T\mathbf{x}_i \sim N(A^T\boldsymbol{\mu}, \Lambda)$, where $\Lambda = diag(\lambda_1, \lambda_2, \ldots \lambda_p)$. The standardized expected shift magnitude along the j^{th} PC can be obtained by $\delta_j = \frac{||\boldsymbol{\mu}|| \cos\theta_j}{\sqrt{\lambda_j}}; j = 1, 2, ..., p$, where θ_i is the angle between the shift direction and the j^{th} PC. As can also be seen from Figure 2.2, this can imply that a PC closer to the shift direction (i.e. smaller θ) will capture a larger shift magnitude. Moreover, if θ is similar for two PCs, then the one with the smaller variance would be more sensitive to the change. Therefore, to take both measures into account, we work with standardized PC score, denoted by $\tilde{y}_{ij} = \frac{y_{ij}}{\sqrt{\lambda_j}}$, that contains both magnitude and sensitivity information.

We choose the EWMA statistic for monitoring as it is more sensitive to small changes and includes the information of previous samples. The EWMA statistic, denoted by z_{ij}, is defined as $z_{ij} = \gamma\tilde{y}_{ij} + (1-\gamma)z_{(i-1)j};\ \ i = 1, 2, ...;\ \ j = 1, 2, ..., p$, where $z_0 = 0$, and $\gamma \in [0, 1]$ is a weight. Under the in-control process, $z_{ij} \sim N\left(0, \sigma^2 = \frac{\gamma}{1-\gamma}\right)$. Consequently, its squared standardized value follows a Chi-squared distribution with one degree of freedom, i.e., $d_{ij} = (\frac{z_{ij}}{\sqrt{\frac{\gamma}{1-\gamma}}})^2 \sim \chi^2_{(1)}$. When the process is out-of-control, depending on the direction of the mean shift, a few d_{ij} values will inflate, while the rest are slightly affected (or unaffected) by the mean shift. To increase the detection power of the monitoring

procedure, these PCs should be filtered out. For this purpose, following [118], we use a soft-thresholding operator to define the following aggregated monitoring statistics,

$$R_i = \sum_{j=1}^{p} (d_{ij} - \nu)_+, \tag{2.1}$$

where the operator $(\cdot)_+ = \max\{0, \cdot\}$, and ν is the threshold value selected based on a desired significance level of χ^2 test. We monitor the R_i statistic and raise an alarm if, $R_i > R_0$, where R_0 is the threshold level found for a desired in-control ARL using simulations.

Selection of Control Limit (R_0***)***

To determine an appropriate value of R_0, we need to know the distribution of monitoring statistic R. We first specify the moments of thresholded values in the 2.2.1.

proposition 2.2.1. *If $d_{i,j} \sim \chi^2_1$, then their soft thresholded values $\tilde{d}_{i,j} = (d_{ij} - \nu)_+$ follows a bimodal truncated χ^2_1 distribution, with the following moments*

$E(\tilde{d}_{ij}) = E((d_{ij} - \nu)_+) = \frac{1}{\Gamma(0.5)}[\Gamma(0.5, \frac{\nu}{2} + e^{-\frac{\nu}{2}}\sqrt{2\nu}] - \nu P(\chi^2_1 > \nu)$

$E(\tilde{d}^2_{ij}) = E((d_{ij} - \nu)^2_+) = \frac{1}{\Gamma(0.5)}[3\Gamma(0.5, \frac{\nu}{2} + e^{-\frac{\nu}{2}}\sqrt{2\nu}(3+\nu)] - 2\nu E(\tilde{d}) - \nu^2 P(\chi^2_1 > \nu)$

Proof provided in the Appendix A.

Now, since we focus on large data streams, i.e., p is large, we can use the Central Limit Theorem to approximate the distribution of R_i. Therefore, as p approaches infinity, the random variables $R_i = \sum_{j=1}^{p}(d_{ij} - \nu)_+$ converge in distribution to a normal distribution $N(\mu_R, \sigma_R)$, where $\mu_R = p\mu_{\tilde{d}}$, and $\sigma_R = \sqrt{p}\sigma_{\tilde{d}}$. Hence, one can use inverse cdf of the standard normal distribution to set R_0 for a desired type I error, α. that is,

$$R_0 = \mu_R + \sigma_R \Phi^{-1}(1 - \alpha), \tag{2.2}$$

where Φ is the inverse cdf of the normal and is the type I error. To validate this approach and the normal approximation, we perform simulations in section 2.3.1. The results show that the empirical α obtained by this approach is very close to the real α.

2.2.3 PC-based Signal Recovery (PCSR) Diagnosis Methodology

In monitoring large data streams, apart from quick detection of changes, precise fault diagnosis to identify accountable variables is extremely crucial [108]. Diagnosis aim is to isolate the shifted variables, which will help in spotting and resolving the root causes of a problem correctly. However, despite its importance, very few diagnostic methods exist for large data streams that can easily be integrated with PCA-based monitoring. Additionally, as mentioned earlier, one of the main drawbacks of PCA-based monitoring is the lack of diagnosability. As PC scores used as monitoring features are linear combinations of original measurements. This is because a PC score is a linear combination of all process variables, which makes the isolation of the altered variables within an out-of-control principal score difficult.

In this section, we propose a diagnostics approach that seamlessly integrates with the proposed PCA-based monitoring for large data streams. Inspired by Compressed Sensing (CS), we propose an adaptive lasso formulation to recover the variables responsible for an out-of-control alarm. We assume that only the process mean has shifted and that the shift is sparse, meaning only a small set of variables changed. In CS, a high-dimensional sparse original signal can be reconstructed from noisy transformed observations by finding solutions to an underdetermined linear system. In other words, given a set of observations y, and a transformation (sensing) matrix Υ, a sparse unknown original signal μ can be recovered from $y = \Upsilon\mu + \epsilon$, where ϵ denotes the random errors.

The outcome of a PC monitoring method can be formulated similarly to identify the shifted process variables. Without loss of generality, we suppose the process has mean $\mathbf{0}$ during in-control that changes to a sparse mean μ during the out-of-control of state. Therefore, the out-of-control observations follow $x = \mu + \epsilon$, where $\epsilon \sim (\mathbf{0}, \Sigma)$. Consequently, the out-of-control PC scores are,

$$y = Ax = A\mu + \tilde{\epsilon}, \tag{2.3}$$

where, $\tilde{\epsilon} = A\epsilon$ is the noise in the PC domain, with zero mean and covariance of Λ= diag$(\lambda_1, \lambda_2, \ldots \lambda_p)$. Looking at Eq. 2.3, we can notice its similarity with a compressed sensing problem. In Eq. 2.3, the eigenvector matrix, A, and the principal scores, $\boldsymbol{y}$, are known, and we wish to estimate the shifted mean μ when an out-of-control situation is detected after monitoring. [18] and [49] showed that a least squares objective function with L_1 penalty, also known as lasso, can be used to estimate the sparse vector μ . Since lasso estimates in general are not consistent, we use adaptive lasso [131] to build our diagnosis model. Specifically,

$$\hat{\mu} = \underset{\mu}{\operatorname{argmin}}\{||\mathbf{y} - A\boldsymbol{\mu}||_{l_2}^2 + \lambda \sum_{j=1}^{p} w_j|\mu_j|, \tag{2.4}$$

where λ is a nonnegative regularization parameter and $\mathbf{w} = \frac{1}{\hat{\mu}_{OLS}}$ is the data dependent weight vector.

One problem in Eq.2.4 is that the covariance matrix ϵ' is not homogeneous. The variance heterogeneity may affect the estimation performance. To address this issue, we apply the following transformation to get constant variances for all error terms.

$$\boldsymbol{y}^* = \Lambda^{-\frac{1}{2}}\boldsymbol{y} \quad , A^* = \Lambda^{-\frac{1}{2}}A \quad , \epsilon^* = \Lambda^{-\frac{1}{2}}\epsilon'. \tag{2.5}$$

Consequently Eq. 2.3 is transformed to$y^* = A^*\mu + \epsilon^*$, where $\epsilon^* \sim (\mathbf{0}, \mathbf{I})$ with $\mathbf{I}$ as a p dimensional identity matrix. The updated adaptive lasso formulation is given by.

$$\hat{\mu} = \underset{\mu}{\operatorname{argmin}}\{||\mathbf{y}^* - A^*\boldsymbol{\mu}||_{l_2}^2 + \lambda \sum_{j=1}^{p} w_j|\mu_j|\} \tag{2.6}$$

Where $w_j = 1/|\hat{\mu}_j|$, and $\hat{\mu}$ is a root p-consistent estimator to μ, e.g., $\hat{\mu} = \hat{\mu}_{ols}$. The optimization problem in Eq. 2.6 can be solved using various optimization algorithms, such as gradient descent, proximal descent, and LARS. In our implementation, we used the gradient descent method. After finding the solution, the set of variables whose corresponding

estimated μ_j is non-zero is considered as the altered variables. It should be noted that according to Theorem 2 in [131], the estimated mean is consistent, loosely meaning that when $\mathbf{A}^*$ has large dimension (i.e. large p), the non-zero components of μ are correctly identified. This implies that the larger the number of data streams is the higher is the likelihood of correct diagnosis. See Appendix B for more details.

To determine the value of parameter λ, one can use Bayesian Information Criterion (BIC) [102] and choose the λ value that results in the smallest BIC value. The reason behind choosing BIC is that it can determine the true sparse model if the true model is included in the candidate set [123]. Since in the diagnosis problem, the objective is to detect the nonzero elements (shifted variables) rather than estimation of the out-of-control mean, BIC is a proper criterion for diagnosis [128].

2.3 Experimental Analysis

In this section, first we validate Proposition 2.2.1 using simulations. Afterward, we study the performance of the proposed monitoring-diagnostic method in change detection and in terms of quick detection of mean shifts and identification of altered variables. For all the experiments, we simulate data streams, such that during an in-control status, they follow a multivariate normal distribution with $N(\mathbf{0}, \Sigma)$, while in out-of-control situations their distribution changes to $N(\mu_1, \Sigma)$, μ_1 is sparse. We carry out the simulations for different levels and types of shifts and the covariance structure and compare the results with existing methods as benchmarks.

2.3.1 Validation of Proposition 2.2.1 for Choosing Control Limits

To validate Proposition 2.2.1, we perform two sets of experiments. In the first experiment, we generate $d_j \sim \chi^2_1$ for $j = 1, \cdots, p$, and $R_i = \sum_{j=1}^{p} (d_{ij} - \nu)_+$ for $i = 1, \cdots, 1000$, similar to Eq. 2.1. Then we calculate R_0 using Eq. 2.2 for desired $\alpha = 0.05$. We calculate the empirical Type I error, denoted by $\tilde{\alpha}$, as the fraction of times R_i's pass the control limit

R_0. We perform this experiment for different values of p and ν and replicate each scenario 1000 times. Finally, we report the average empirical Type I errors in table 2.1.

In the second experiment, first we simulate $\mathbf{x}_i$ for $i = 1, \cdots, 1000$ as a p dimensional normal distribution random variables with random covariance matrix and zero mean. Given the eigenvector matrix $\mathbf{A}$, we calculate its PC scores y_{ij} , and its corresponding EWMA statistic is calculated as z_{ij} using $\gamma = 0.4$. Consequently, its squared standardized value are calculated as $d_{ij} = (\frac{z_{ij}}{\sqrt{\frac{\gamma}{1-\gamma}}})^2$. Here, for each observation we define $R_i = \sum_{j=1}^{p}(d_{i,j} - \nu)_+$, we repeat this procedure 1000 times. Similar to previous experiment, we calculate R_0 using Eq. 2.2 for desired $\alpha = 0.05$. We calculate the empirical Type I error, $\tilde{\alpha}$, as the fraction of times R_k's pass the control limit R_0. We perform this experiment for different values of p and ν and replicate each scenario 1000 times. Finally, we report the average empirical type I errors in table 2.2.

As can be seen from both tables, as p increases, the empirical Type I error approaches to its true value $\alpha = 0.05$. Moreover, for large p, the result is less sensitive to the choice of the threshold value (ν). Hence, it shows the validity of the proposed approach for finding control limits. Note that the main difference between these studies is the independency of R_is. In the first study, R_is are independently generated, whereas in the second study, R_is are calculated using EWMA statistics, which are not independent. The larger bias in the results of the second study is mainly because that the monitoring statistic values are autocorrelated. However, for very large p (e.g, $p > 5000$), this difference is negligible. For smaller p, we would suggest using a Monte Carlo simulation to determine the control limit.

Table 2.1: Empirical type I error using first experiment

		p				
		100	500	1000	5000	10000
ν	0.05	0.0090	0.0067	0.0063	0.0056	0.0053
	0.10	0.0091	0.0067	0.0060	0.0055	0.0052
	0.15	0.0090	0.0067	0.0062	0.0056	0.0054
	0.20	0.0090	0.0066	0.0064	0.0055	0.0054
	0.25	0.0091	0.0068	0.0061	0.0055	0.0054
	0.35	0.0092	0.0068	0.0063	0.0055	0.0053

Table 2.2: Empirical type I error using second experiment

		p				
		100	500	1000	5000	10000
ν	0.05	0.0091	0.0071	0.0068	0.0065	0.0050
	0.10	0.0091	0.0073	0.0067	0.0064	0.0050
	0.15	0.0090	0.0073	0.0067	0.0063	0.0050
	0.20	0.0091	0.0072	0.0067	0.0063	0.0051
	0.25	0.0089	0.0072	0.0068	0.0061	0.0049
	0.35	0.0083	0.0066	0.0063	0.0057	0.0047

2.3.2 Monitoring Methods Analysis

In this section, we conduct various simulations to validate the performance of the proposed monitoring method based on the Average Run Length (ARL) and its standard error for different magnitudes of shifts. Specifically, the following scenarios are considered:

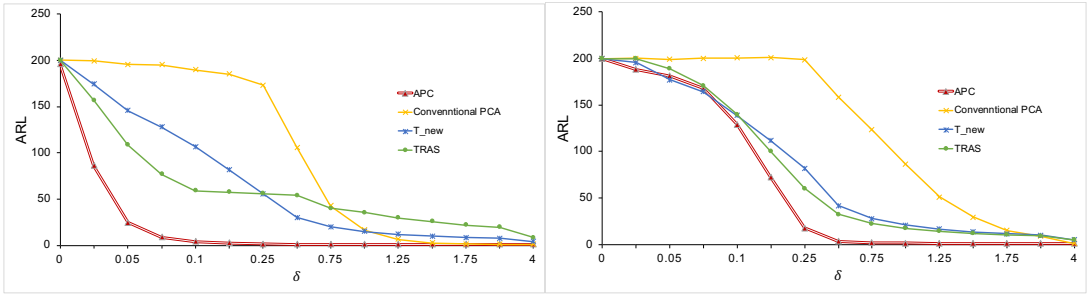
I) Random covariance structure and random shift: To generate the random covariance matrix, we use the Wishart distribution with diagonal entries equal to 1. To generate a sparse mean shift, we randomly select 20% of the process variables and shift them by δ.

II) Block diagonal covariance: This scenario mimics the situations where each data stream is correlated with only a subset of the data streams. The covariance matrix used in this scenario has $K = 12$ blocks, denoted as B_k, $k = 1, ..., K$, where each block B_k is a random semi-positive definite matrix generated from a Wishart distribution. To generate out-of-control data, we shift the mean of a set of variables that belong to only one of the blocks (B_k), by δ , i.e., $\mu_j = \begin{cases} \delta & j \in B_k \\ 0 & j \notin B_k \end{cases}$.

In each scenario, p data streams are generated. We run the simulations for, $p = 100$, $p = 1000$, and $p = 10,000$ to evaluate the performances in different dimensions. We apply the proposed monitoring method, APC, and compare it with three existing methods: a)

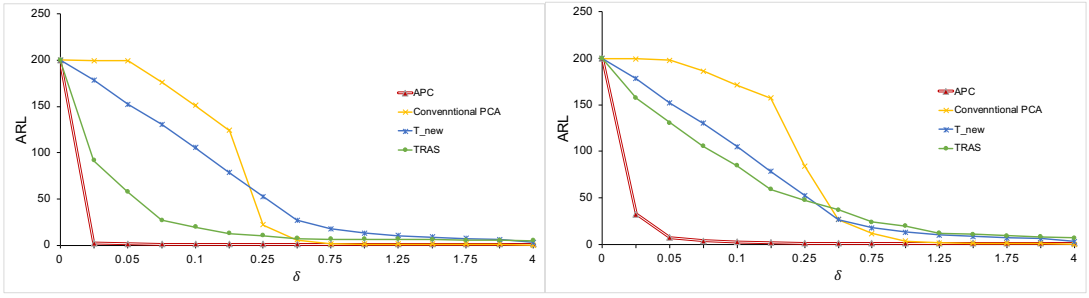
T_{new} [130]: This monitoring method is based on a goodness-of-fit test of the local CUSUM statistics from each data stream, b) TRAS [77]: Top-r based adaptive sampling (TRAS) is an adaptive sampling strategy that uses the sum of top-r local statistics for monitoring. Since this method works only for independent variables. We will implement it on PCs rather than original data, c) Traditional PCA-based monitoring: In this approach, the selected number of components to retain in the model is based on the cumulative percentage of variance (CPV) equal to 90. Control charts are constructed by using the Hotelling's T^2 statistic and the Q statistic. [25]

To detect an out-of-control condition, the control limits are set such that the ARL for in-control observations is equal to 200 (this corresponds to a significance level of 0.005). Each control limit is calculated through 1000 replications. The results are shown in Figures 2.5,2.6, and 2.7. For $p = 100$, as shown in Figure 2.5(a), the proposed APC markedly outperforms the other benchmark methods. Even for shifts as small as $\delta = 0.1\sigma$, the ARL for APC is 4.06. This is about fourteen times smaller than the second best method, which is TRAS with ARL equal to 56. Moreover, for shifts $\delta \geq 0.1\sigma$, APC detects the shift almost instantly (i.e., $ARL_1 = 1$). The results for Scenario II, shown in Figure 2.5(b), also show that APC is superior to others, especially for moderate and large shifts. As an example, for a shift with the magnitude of $\delta = 0.25\sigma$, APC's ARL is 17.67, while this values for the best benchmark (TRAS) is 61.37. As expected, the out-of-control ARL values for all methods in Scenario II is larger than those in Scenario I. For higher dimensions, the APC's ability to detect shifts becomes even better while the other method's performances stay the same or deteriorate. This shows that as dimension grows, the shifts are easier to be captured in PC scores that are in the direction of the shift. Therefore, this study indicates that the APC method outperforms other methods for detecting small values of shifts. Also, as dimension grows APC is a clear choice for prompt change point detection. This is due to the adaptive nature of the proposed monitoring statistic.



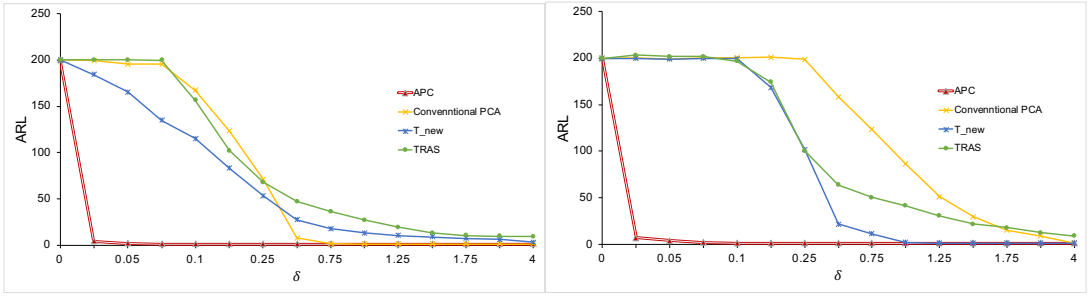
(a) scenario I (b) scenario II

Figure 2.5: ARL of scenarios I, II for different values of δ (shift magnitude) for $p = 100$



(a) scenario I (b) scenario II

Figure 2.6: ARL of scenarios I, II for different values of δ (shift magnitude) for $p = 1000$



(a) scenario I (b) scenario II

Figure 2.7: ARL of scenarios I, II for different values of δ (shift magnitude) for $p = 10,000$

2.3.3 Diagnosis Analysis

In this section, In addition to scenarios I and II presented in the previous section, we add another scenario (scenario III) with an autoregressive covariance matrix, i.e., $\rho_{ij} = |0.5|^{(i-j)}$ for variables i, j. We validate the performance of the proposed diagnosis method for different numbers of shifted variables (PS) as well as their shift magnitudes (δ), using the following performance measures: (a) False negative percentage (%FN), defined as the percentage of the number of variables that are not detected over the number of all faulty variables; (b) False Positive percentage (%FP), defined as the percentage of the ratio of number of variables that are mistakenly detected as faulty over the number of all not-faulty variables; (c) parameter selection score (PSS), defined as the total number of variables that are labeled incorrectly (either as faulty or not-faulty); and (d) F1-Score, defined as the harmonic average of the precision and recall, and indicates our overall performance in terms of FP and FN and ranges from 0 to 1. For FN, FP and PSS measures, the smaller the value, the better the performance is, whereas for F1-score the higher the better. We compare the performance of our proposed method with the Adaptive Lasso-based approach proposed by [128] which is an approach for diagnosis of sparse changes. The results for $p = 100$ are shown in Tables 2.3, 2.4, and 2.5, and Figures 2.8, 2.9, and 2.10 for scenarios I, II, and III respectively. In these tables PS denotes the percentage of shifted process variables. As shown in Table 2.3 and Figure 2.8, under scenario I, when shift occurs in a random set of variables with a random covariance matrix, our proposed PCSR performs better than the LEB for most of the cases except for the case with PS=10% and small shifts (i.e., $\delta = 0.7\sigma$) where LEB is slightly better (but very close) than our method. However, For larger shifts, PCSR outperforms the LEB method. For instance, for a shift equal to 1σ and percentage of shifted variables equal to 10% the $F1$ accuracy using PCSR is equal to 0.9881 while it is equal to 0.9363 for LEB method. In Scenario II, as can be seen from the table and figure, PCSR clearly outperforms the LEB method. For example for a shift equal to 0.7σ on 10% of variables, PCSR's F1-score is 0.6802 while, the LEB F1 score is 0.3725. Also, Table 2.5

and Figure 2.10 present the comparison of methods based on scenario III. The results of our simulation in scenario III indicate the superior performance of that our method. For instance, for a 0.5σ shift on 25% of variables, PCSR's F1-score is 0.7173 while, this value for LEB is 0.4648. This results shows that for random non-sparse covariance matrices, the LEB method and PCSR method performs almost similarly. However, for sparse covariance matrices such as a block covariance or an autoregressive covariance, PCSR clearly outperforms LEB method. These sparse occurance of covariance matrices are very common in real world. This is because of the fact that in many situations, each data stream is correlated only with a small group of other data streams, but is not correlated with all other data streams collected in the system. Hence, a method that can detect the changes in such systems is necessary and more appropriate for real-world applications.

In short, the results of the simulation study not only show the effectiveness of our method in identifying the set of altered variables, but also indicate the superiority of our method to the benchmark. This implies that the proposed method can address the long-standing issue with PCA-based monitoring due to the lack of diagnosability of an out-of-control alarm.

2.4 Case Study

In this section, we apply the proposed monitoring and diagnosis methods to two case studies on a) defect detection in a steel rolling process, and b) quality monitoring of wine. Additionally, we compare our results with existing methods.

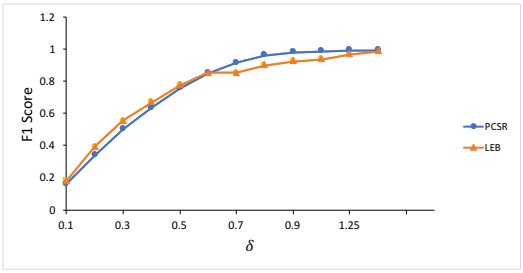
2.4.1 Defect detection in Steel Rolling Process

Early detection of process shifts in a rolling process is necessary to avoid damage to products and reduce manufacturing costs. Rolling is a high-speed process that makes its monitoring particularly challenging. In this study, we show that how the PCA-based method can effectively detect anomalies and damages imprinted on a steel bar after rolling. The

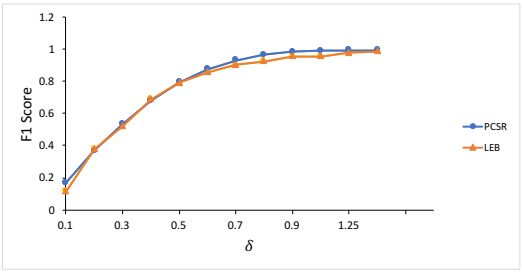
Table 2.3: Diagnosis simulation results for Scenario I

	Shift	PCSR				LEB			
		FP%	**FN%**	**PSS**	**F1**	**FP%**	**FN%**	**PSS**	**F1**
PS=0.1	0.1	86.99	5.289	13.459	0.1603	85.86	4.28	12.438	**0.1790**
	0.3	41.36	8.652	11.923	0.5001	38.11	6.757	9.891	**0.5554**
	0.5	8.05	6.107	6.301	0.757	6.56	5.501	5.607	**0.7763**
	0.7	0.62	2.272	2.107	**0.9141**	0.62	3.924	3.594	0.8538
	1	0	0.29	0.261	**0.9881**	0	1.580	1.422	0.9363
	1.25	0	0.1567	0.141	**0.9934**	0	0.761	0.685	0.968
	1.5	0	0.156	0.14	**0.9935**	0	0.347	0.312	0.9850
PS=0.15	0.1	87.267	6.388	18.52	**0.1672**	92.067	3.466	16.756	0.1158
	0.3	42.773	10.231	15.112	**0.5323**	45.84	9.379	14.848	0.520
	0.5	9.16	7.029	7.349	**0.7927**	8.853	7.077	7.343	0.7916
	0.7	0.687	2.68	2.381	**0.9310**	0.433	3.914	3.392	0.9015
	1	0	0.364	0.309	**0.9903**	0	1.711	1.454	0.9552
	1.25	0	0.242	0.206	**0.9935**	0	0.854	0.726	0.9772
	1.5	0	0.229	0.195	**0.9938**	0	0.439	0.373	0.9882
PS=0.25	0.1	86.676	7.303	27.146	**0.1941**	88.632	4.815	25.769	0.1769
	0.3	48.268	10.277	19.775	**0.5653**	52.116	8.736	19.581	0.5460
	0.5	15.896	6.939	9.178	**0.8208**	19.848	6.372	9.741	0.8028
	0.7	1.228	3.123	2.649	**0.9505**	3.004	6.123	5.343	0.9018
	1	0	0.489	0.367	**0.9929**	0	3.145	2.359	0.956
	1.25	0	0.371	0.278	**0.9946**	0	1.876	1.407	0.9732
	1.5	0	0.368	0.276	**0.9947**	0	1.161	0.871	0.9832

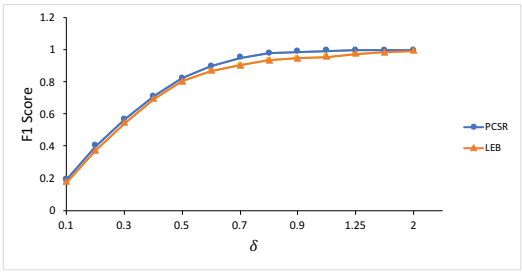
dataset we consider here, includes images of size 128×512 pixels of the surface of rolled bars collected by a high-speed camera [122]. Of the 100 images, the first 50 images are in-control. One example of the image of rolling data for in-control vs out-of-control process is shown in Figure 2.11. We use this data to simulate an image with in-control observations in the first 126 rows and out-of-control observations in the remaining 72 rows. The generated image is presented in Figure 2.12. Also, we crop the image at the right end to avoid the non-informative dark segment of the image. Hence, our generated picture is of the size of 198×300. In this study, each row of an image (a vector of 300×1) is treated as an observation, creating a multi-stream data with the size of 300. As the picture shows, for out-of-control observations, some small black lines, indicating anomalies, emerge at the left part of the frame. We are interested to see whether our monitoring approach can detect



(a) PS=0.10

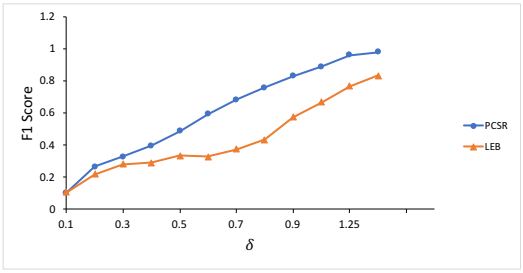


(b) PS=0.15

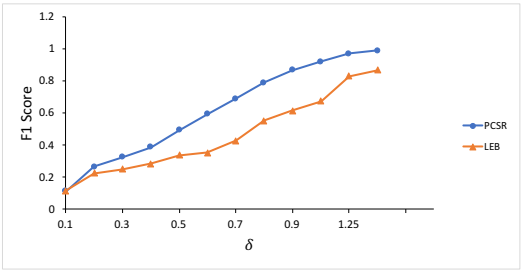


(c) PS=0.25

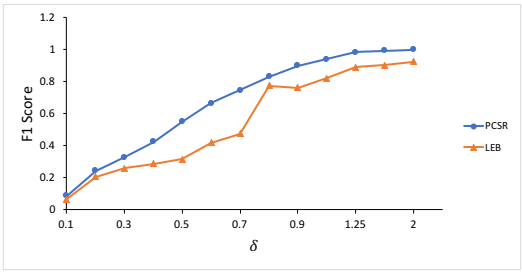
Figure 2.8: F1 of scenarios I different values of δ (shift magnitude)



(a) PS=0.10



(b) PS=0.15



(c) PS=0.25

Figure 2.9: F1 of scenarios II different values of δ (shift magnitude)

Table 2.4: Diagnosis simulation results for Scenario II

PS	Shift	PCSR				LEB			
		%FP	**%FN**	**PSS**	**F1**	**%FP**	**%FN**	**PSS**	**F1**
PS=0.10	0.1	93.775	0.40435	7.874	0.1011	93.663	0.91304	8.333	**0.1049**
	0.3	78.412	0.5337	6.764	**0.3257**	81.562	0.93478	7.385	0.2784
	0.5	63.1	0.75652	5.744	**0.4854**	76.987	0.81848	6.912	0.3333
	0.7	40.387	1.0446	4.192	**0.6802**	72.162	0.96848	6.664	0.3725
	1	8.575	1.2109	1.8	**0.8875**	33.562	2.4522	4.941	0.6672
	1.25	0.7375	0.71413	0.716	**0.9586**	12.425	3.5022	4.216	0.7664
	1.5	0	0.40217	0.37	**0.9789**	0.6625	3.612	3.376	0.8302
PS=0.15	0.1	93.537	0.50714	15.392	0.1119	93.419	0.75476	15.581	**0.1160**
	0.3	79.588	0.69405	13.317	**0.3225**	84.9	1.1179	14.523	0.2475
	0.5	64.444	0.98333	11.137	**0.4925**	77.825	1.3536	13.589	0.3355
	0.7	42.219	1.3167	7.861	**0.6891**	68.863	1.7393	12.479	0.4258
	1	7.8375	1.519	2.53	**0.9182**	39.062	2.9917	8.763	0.6712
	1.25	0.81875	1.0476	1.011	**0.9696**	7.0625	5.8714	6.062	0.8282
	1.5	0.00625	0.52262	0.44	**0.9869**	0.225	5.969	5.05	0.8662
PS=0.25	0.1	95.088	0.56447	23.25	**0.0878**	96.525	0.87105	23.828	0.0641
	0.3	79.483	1.0329	19.861	**0.3251**	84.133	1.5408	21.363	0.2593
	0.5	59.008	1.7592	15.499	**0.5489**	80.083	1.3763	20.266	0.3140
	0.7	36.492	1.7855	10.115	**0.7440**	64.933	2.2645	17.305	0.4742
	1	6.4167	1.8013	2.909	**0.9379**	15.758	6.4645	8.695	0.8180
	1.25	0.57083	1.1645	1.022	**0.9792**	2.625	7.0224	5.967	0.8878
	1.5	0.0083333	0.62895	0.48	**0.9903**	0.1	7.1421	5.452	0.8993

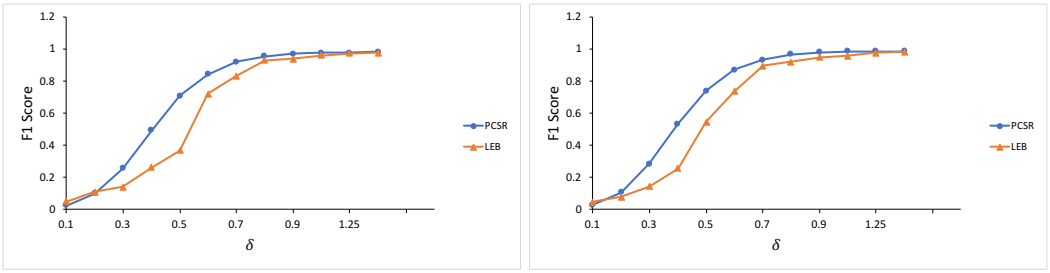
this change, and whether the diagnosis approach can determine the changed pixels.

We apply our proposed APC method for monitoring the process. We use the first 70 in-control data (the first 70 rows of the image) to obtain the control limits. The control limits are determined according to the procedure explained in Sec. 2.3.2 to achieve the in-control ARL of 200. The resulting control chart is shown in Figure 2.13. As can be seen from the figure, after the change point, our monitoring statistic instantly inflates and raises an out-of-control alarm by the first observation after the change. Furthermore, to compare its performance with existing state-of-the-art techniques, we report the run lengths (the number of observations before the change is detected) for each method in Table. 2.6. We can see that APC has the smallest RL, hence fastest in detecting the change in comparison with other benchmarks.

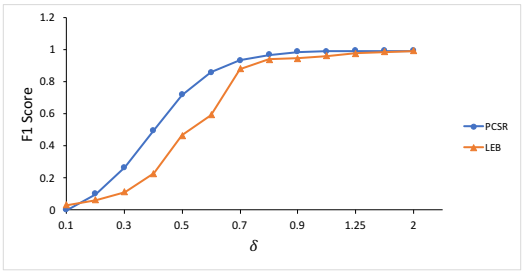
Table 2.5: Diagnosis simulation results for Scenario III

	Shift	PCSR				LEB			
		%FP	**%FN**	**PSS**	**F1**	**%FP**	**%FN**	**PSS**	**F1**
PS=0.10	0.1	98.48	0.63667	10.421	0.0257	97.31	0.87222	10.516	**0.0484**
	0.3	82.78	0.85778	9.05	**0.2562**	92.08	0.47667	9.637	0.1385
	0.5	38.18	1.1511	4.854	**0.7073**	73.69	0.40889	7.737	0.3675
	0.7	7.44	1.0178	1.66	**0.9176**	17.54	1.4233	3.035	0.8303
	1	0.11	0.61556	0.565	**0.9740**	0.38	0.94444	0.888	0.9595
	1.25	0	0.60111	0.541	**0.9752**	0.02	0.67	0.605	0.9722
	1.5	0	0.46889	0.422	**0.9806**	0	0.62444	0.562	0.9742
PS=0.15	0.1	98.347	0.6259	15.284	0.0293	97.513	0.84824	15.348	**0.0460**
	0.3	81.48	1.0471	13.112	**0.2832**	91.853	0.31059	14.042	0.1431
	0.5	35.48	1.4965	6.594	**0.7390**	56.067	0.77882	9.072	0.5457
	0.7	6.5533	1.2553	2.05	**0.9319**	10.033	1.9471	3.16	0.8924
	1	0.11333	0.64706	0.567	**0.9821**	0.19333	1.6247	1.41	0.9567
	1.25	0	0.60824	0.517	**0.9837**	0	0.96471	0.82	0.9744
	1.5	0	0.60941	0.518	**0.9836**	0	0.68	0.578	0.9818
PS=0.25	0.1	98.564	0.64533	25.125	0.0264	98.312	0.86	25.223	**0.0323**
	0.3	83.508	1.184	21.765	**0.2641**	93.74	0.272	23.639	0.1128
	0.5	40.056	1.86	11.409	**0.7173**	63.888	1.124	16.815	0.4648
	0.7	8.856	1.604	3.417	**0.9292**	13.84	3.1027	5.787	0.8776
	1	0.184	0.78	0.631	**0.9878**	0.348	2.7707	2.165	0.9593
	1.25	0.004	0.65333	0.491	**0.9905**	0.008	1.8747	1.408	0.9732
	1.5	0	0.60133	0.451	**0.9913**	0	1.1693	0.877	0.9831

Diagnosis. To check the performance of our PCSR method, we performed diagnosis using our method vs LEB method on the out of control data. The 70 phase 1 data are used as the ground truth, and 25 out-of-control observations are used to detect the changed pixels in the generated image. The area selected as out-of-control for each method as well as the in-control and out-of-control images are shown in Figure 2.14. The identified pixels are shown in black and the remaining unchanged pixels are shown in white in Figure 2.14(c) and (d), respectively. As the results show, PCSR method clearly detects the changed pixels in the image with no false detection. Note that although LEB can identify the changed pixels, it generates a few false detection areas.



(a) PS=0.10 (b) PS=0.15



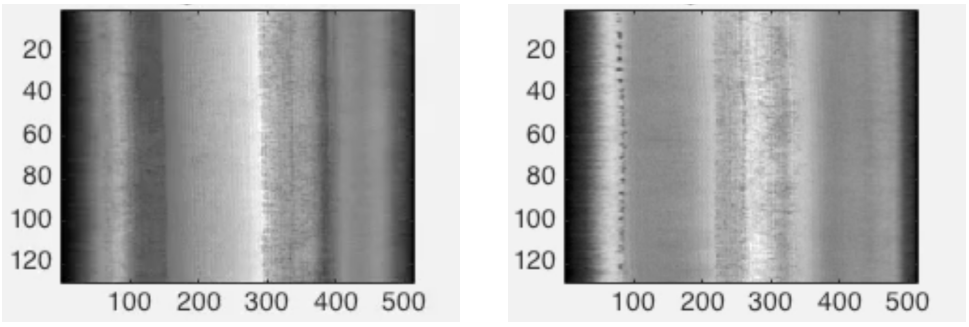
(c) PS=0.25

Figure 2.10: F1 of scenarios III different values of δ (shift magnitude)

2.4.2 Wine Quality Monitoring

In this section, to show the effectiveness of our approach, we apply our proposed methodology on white wine production process. This dataset is a real dataset that we collect from the UCI data repository [1]. The data has 4898 observations obtained between May 2004 to February 2007 for the purpose of enhancing the quality of Portuguese Vinho Verde wine.

[1] http://archive.ics.uci.edu/ml/datasets/Wine+Quality



(a) Rolling image at time 1 (b) Rolling image at time 90

Figure 2.11: Image of rolling data for in control process (a) and out of control process (b)

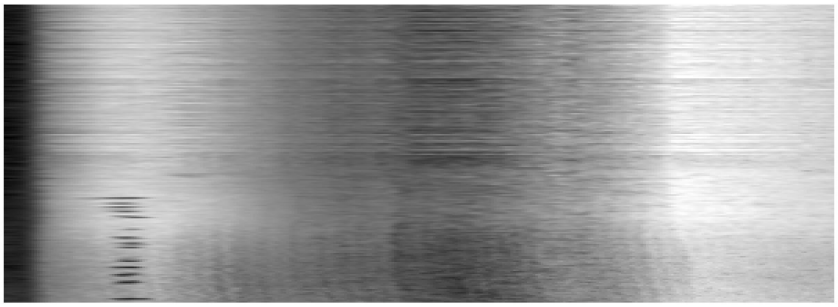


Figure 2.12: Generated Image with first 126 rows as in-control and remaining 72 rows as out-of-control

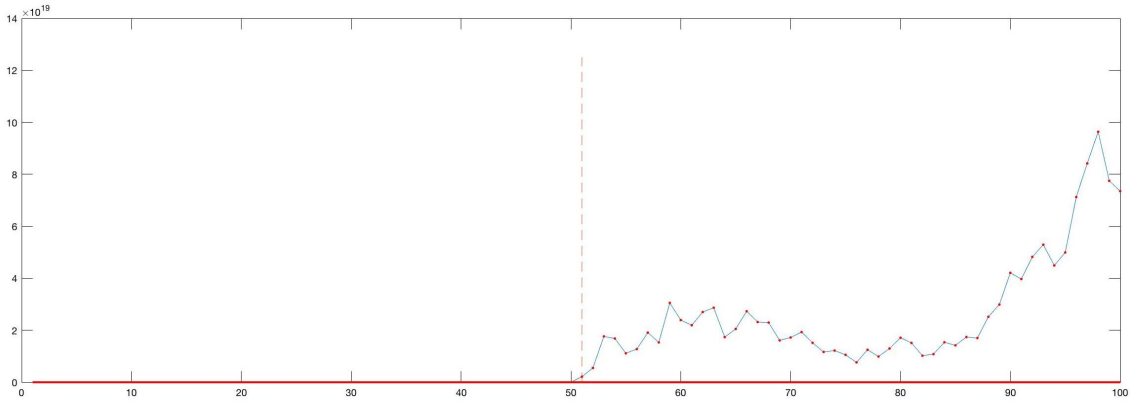
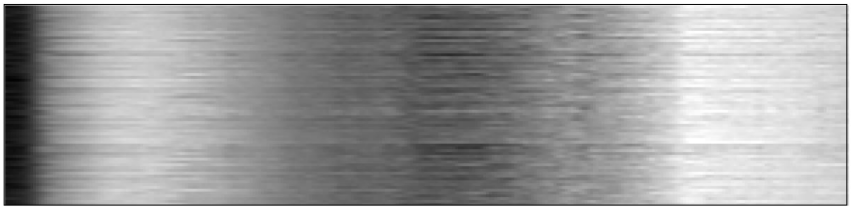
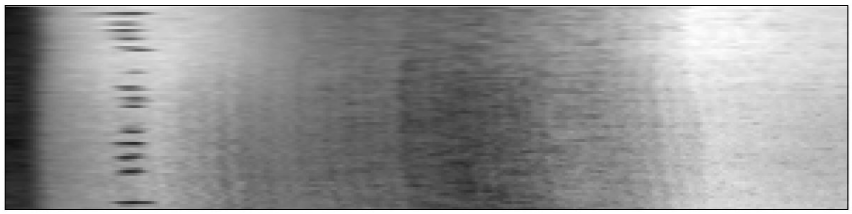


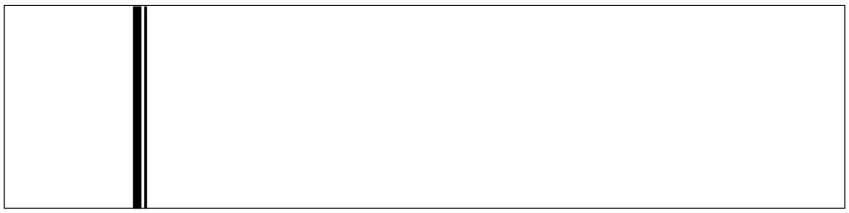
Figure 2.13: Monitoring Rolling Data using APC Method



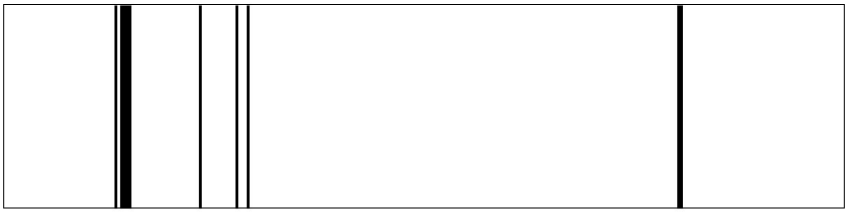
(a) In Control Image



(b) Out of control image



(c) Diagnosis Image Using PCSR



(d) Diagnosis Image Using LEB

Figure 2.14: Diagnosis using PCSR and LEB method)

Table 2.6: Run length Comparison of different methods in detecting the change point

Method	**Detected Change Point**
APC	**1**
Conventional PCA	16
T_new ([130])	10
TRAS ([77])	14

The collected data has eleven variables named as fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, PH, sulphates and alcohol. Additionally, an additional (manually annotated) quality variable is available that will be used as ground truth for the wine quality. This variable ranges between 0 (low quality wine) and 10 (high quality wine) that is gathered based on sensory analysis [23].

Our objective in this case study is to monitor the wine quality using the variables and diagnose the shifted variables, if there is a shift. We perform the APC study on this dataset. Similarly to [130], we focus on a subset of the data in which the quality variable is either 6 or 7. The observations with the quality of 7 are considered as acceptable while the rest are unacceptable, hence out-of-control. The quality variable will be used to gauge the performance of our monitoring—the monitoring should raise an out-of-control alarm as soon as the quality variable is going down from 7 to 6. When the alarm is raised, our diagnosis approach should be able to pinpoint the actual shifted process variables.

Overall there are 880 observations with the quality equal to 7 of which 830 observations are used for phase I monitoring. Also, we set the control limits (R_0 in APC method) such that ARL for in-control observation is 1000. To do the comparison, we implement our method along with the existing methods shown in Sec. 2.3.2. All parameters in the methods are set to achieve in-control ARL of 1000 so that methods are comparable.

For phase II monitoring, we use the remaining 40 points with the quality of 7 followed by observations with the quality of 6. The goal is to investigate how fast and accurately our monitoring algorithm detects the change point in comparison to the existing methods.

The results are shown in Figure 2.15. and Table 2.7. As shown in Table 2.7, the APC

Table 2.7: Comparison of different methods in detecting the change

Method	N_A=Number of observations after Chang point until alarm
APC	**11**
Conventional PCA	23
T_new [130]	24
TRAS [77]	28

monitoring method detects the change after 11 observations. On the other hand, as shown in the table, other methods took more than twice as many observations to detect the change.

Monitoring:

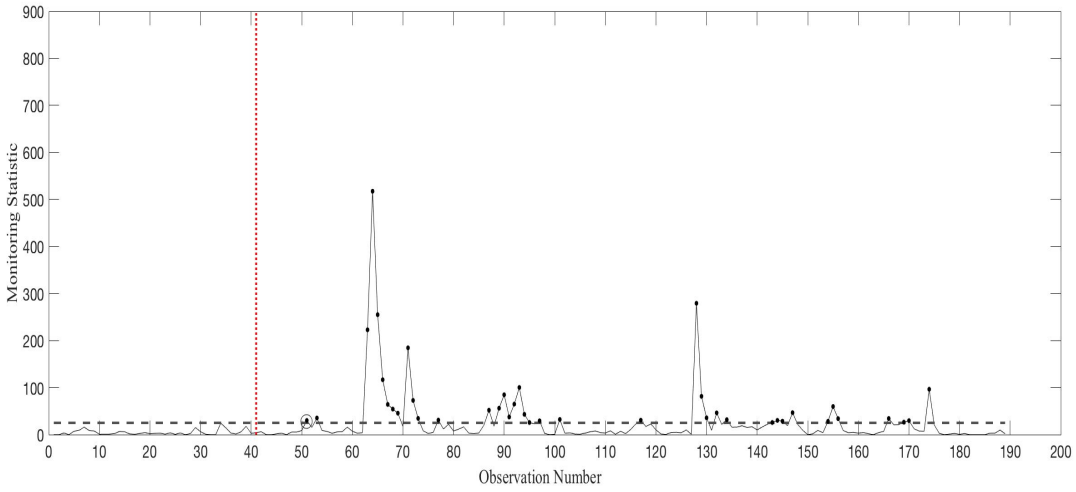


Figure 2.15: Monitoring Wine Quality Data using APC Method

Diagnosis. PCSR diagnosis approach, among the eleven variables, four were determined as shifted variables. The detected shifted variables using our method are residual sugar, chlorides, density, and alcohol. Using the LEB method discussed in Sec. 2.3 the variables chlorides, density and alcohol were selected as well.

2.5 Conclusions

In this paper, we proposed an SPC framework for high-dimensional data streams that seamlessly integrates monitoring and diagnostics. We proposed a new PCA-based monitoring

approaches, viz. Adaptive PC Selection (APC) monitoring. We first negated the common belief that the high-PCs (principal components with highest variances) should be used for as monitoring features, and then, showed that monitoring adaptively selected PCs will be more effective. Using simulations, we showed that adaptively selected PCs outperforms other benchmark methods for different types of covariance matrix structures and types of shifts. Importantly, in all scenarios, the conventional approach of monitoring high-PC was shown to have poorer performance.

In the diagnosis module, we first discussed the challenge in finding the shifted variables after a PCA-based monitoring procedure. The challenge lies in isolating the process variable whose linear combination results in the signaling PC. We used the CS principle to formulate an adaptive Lasso estimation of a subset of altered variables. This formulation takes the eigenvectors and principal components (after a shift) as inputs and yields the process variables that caused the shift. Our experimental validations showed that PCSR performs markedly better than the benchmark method.

Furthermore, we showed the practical applicability and validity of our methods via real-world case studies. The first case study was on defect detection in a steel rolling process, in which we found that the proposed APC detects the shift faster than all the other methods. Moreover, the PCSR diagnosis approach detects the change pixels better than the existing method. In another case study, we monitored wine quality and diagnosed the shift. Our monitoring approach was again faster and our diagnosis approach could find an additional shifted process variable, *sugar*, that was missed by the existing diagnostics approach. In this paper, we have focused on monitoring and diagnosing the mean shifts. While the developed APC can potentially be used to detect shifts in covariances, further research is required to extend the PCSR diagnostics approach to the covariance matrix monitoring.

CHAPTER 3
DYNAMIC NETWORK MONITORING USING EXTENDED KALMAN FILTER ON HURDLE MODELS

3.1 Introduction

As a consequence of the 2007 financial crisis, according to the U.S. Financial Crisis Inquiry Commission's final report in January 2011, 8.5 million families lost their homes in foreclosure or were severely behind on their mortgage payments [35]. The unemployment rate peaked at about 10 percent in October 2009 [116], and the stock market suffered record losses, with the S&P 500 Index losing 55 percent of its value between October 2007 and March 2009 [99]. Nearly half a trillion dollars of taxpayer money was spent to stabilize the financial economic system [115]. Indeed, the financial crisis induced large societal costs in the form of slower economic growth and direct bailouts, and has thus clearly accentuated the need for more effective monitoring and oversight of financial markets and institutions.

Researchers have responded to this call by developing new methods and techniques to capture the interconnectedness among financial institutions. Measuring interconnectedness is crucial because it can contribute to crises by amplifying the effects of negative events. For example, a break down in interconnectivity within lending markets for banks (i.e., banks stop lending to each other) has been shown to intensify the impact of small negative economic shocks and result in financial crises [64] through traditional bank runs [16] and other mechanisms. Besides, it is difficult for individual firms or even a regulatory body that has access to broader information to assess if and when interconnectivity has shifted to a new, dangerous state that can contribute to systemic events. It is here that financial network analysis and statistical process control offer a promising solution and also the central focus of our work.

A financial network describes a collection of financial institutions (nodes) and the links between them. Edges in financial networks reveal information about the underlying balance sheets of the connected firms. Thus, the main idea in financial network analysis is to draw insights about the level of systemic risk from connectivity patterns, e.g., a sparsely connected interbank lending network can indicate that banks have stopped participating in the interbank market due to greater perceived counter-party risk, which has systemic risk implications [36, 15]. Supporting the notion of monitoring financial networks over time for risk management, multiple works have shown that network statistics, such as the average of the network degree distribution, can shift depending on stable or crisis market conditions [13, 36, 15, 26, 7, 8].

While the literature has established the importance of network statistics for early warning systems, to our knowledge explicit methodology to *systematically* identify in real-time whether the network has entered a new epoch has not yet been developed. This is an especially significant problem in practice given that high false positive rates that can occur unless a careful approach is utilized that can distinguish gradual change resulting from the typical edge dynamics from abrupt changes in trading patterns underlying the financial system. Here we address this gap in the literature by demonstrating a new methodology to detect change points within a sequence of sparse financial networks with additional node and edge information. Specifically, we monitor networks on the e-MID trading platform, the only electronic regulated interbank market in the world, from January 2006 to December 2012. Edges are defined by the number of overnight loans between European banks on this platform, that is, if Bank A lends to Bank B, then an edge is drawn from Bank A to Bank B and weighted by the number of directed loans in the given week.

The main idea behind our approach is to use a state space model to capture the temporal dynamics of the edge formation process, which is modeled as a function of nodes and edges attributes such as whether two banks are originating in the same country. Specifically, we model the number of loans between banks using the Hurdle model [88], which uses a

Figure 3.1: Timeline of estimation results and main events in the financial crisis. The proposed monitoring framework would have raised alarms in real time to regulators about changes in interbank market conditions that coincide with the onset and end of the crisis.

stochastic process to capture a critical initial decision that banks make about whether to participate in or exit a financial market in addition to modeling the number of interactions between banks in a second stage. The Extended Kalman Filter [78] is used as an online, recursive inference procedure to estimate and update the regression parameters over time. Finally, we generate a one-step-ahead prediction of the network and compare it to the realized network to decide whether the observed evolution was smooth or abrupt using an Exponentially Weighted Moving Average (EWMA) control chart. To our knowledge the combination of the Hurdle model with the Extended Kalman Filter to monitor sparse sequences of networks is a novel methodological contribution.

As shown in Figure 3.1, using our monitoring approach on the e-MID data, a regulator could raise the alarm in real time about the change from calm to crisis conditions during the week of June 18, 2007. Note that this is well before August 8-9, 2007, which is widely considered the official recognition of the crisis when central banks around the world announced major liquidity shortages [15]. This is a notable finding given that it is difficult to correctly identify the onset of the crisis from typical financial variables in our data (see Figure 3.2) and that previous research using network analysis on the same data had difficulty correctly identifying this moment as the beginning of the crisis. Indeed, [36] conduct

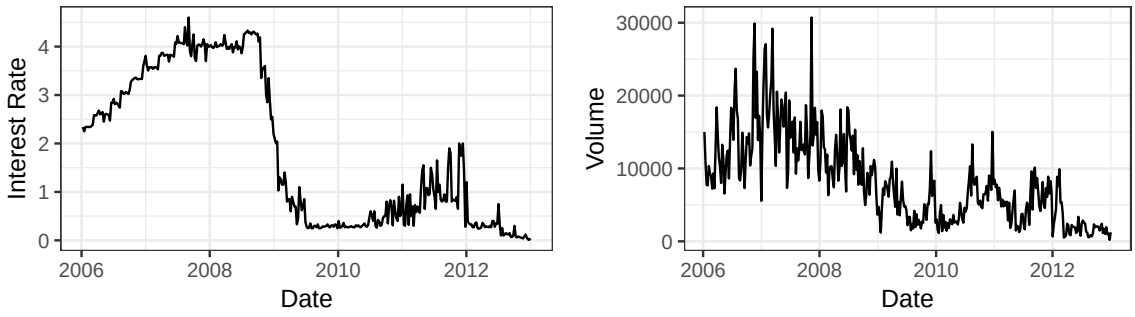


Figure 3.2: Weekly interest rate and volume in the e-MID interbank market.

a detailed network analysis of data from the same market and remark that "The start of the GFC [global finance crisis] is not easy to determine [from the data]...."

In addition to the online monitoring, our methodology can also be used to identify the exact date of the change point for root-cause analysis. We find a change to crisis conditions in the interbank market dated to the week of March 12-16, 2007, which coincides closely with a Freddie Mac announcement on February 27, 2007, that they would no longer buy the riskiest type of mortgages (sub-prime), the root-cause event as identified by the [33]. Similarly, our methodology successfully detects in real-time the end of the crisis before official announcements as such by central banks, and identifies the change point date marking the end of the crisis as October 6-10, 2008, which coincides with the so-called bank bailouts (the Emergency Economic Stabilization Act of 2008; [114]) being signed into law by then President Bush on October 3, 2008.

Thus, we find promising results that demonstrate the proposed approach could be a valuable tool for regulators to utilize when monitoring the financial system. We also study our approach through simulation and find consistently positive results underscoring the generality of the posited approach for the financial context as well as other settings where dynamic networks are encountered, such as traffic networks [22], co-citation networks [21, 59], social networks [105, 65], gene regulatory networks [127], among many other areas [103, 34].

The next section presents further details about the data, followed by the proposed mod-

els and estimation framework in Section 3.3. Through a detailed simulation study benchmarking different monitoring approaches (Section 3.4) as well as with the real data (Section 3.5), we show that the proposed model performs favorably when compared to competing methods for monitoring sparse network connectivity. The paper concludes with a short discussion on the overall findings, the limitations of our work, and directions for future research in Section 3.6.

3.2 Data

The e-MID market is open to all banks admitted to operate in the European interbank market. In August 2011, e-MID had 192 banks from European Union countries and the U.S., including 29 central banks that performed as market observers [36]. Our data contains all e-MID transactions from January 2006 through December 2012. Each transaction includes the date, lender, and borrower (with their real names anonymized), country of origin for lender and borrower, interest rate, quantity, and an indication of which party initiated the trade. The data includes 40-60 banks that are publicly-traded.[1] For these banks, we also observe their weekly returns in the stock market.

Figure 3.2 shows weekly volume and interest rates in the e-MID market. As the financial crisis progressed, interest rates dropped in level to near zero and activity in the market decreased markedly. In fact, the changes in these financial variables reflect major real-world events. For example, using the same data, [15] analyze four sub-periods: (1) a pre-crisis period from January 2, 2006, until August 7, 2007, when the European Central Bank (ECB) noted worldwide liquidity shortages; (2) the first crisis period from August 8, 2007 until September 12, 2008, when Lehman Brother's collapsed; (3) the second crisis period from September 16, 2008, through April 1, 2009 when the ECB announced the end of the recession; and (4) post-recession period, from April 2, 2009, onwards. Similarly, [36] write "The start of the GFC [global finance crisis] is not easy to determine [from the

[1]The exact number is not given to protect confidentiality. Bank identities in the interbank market are confidential.

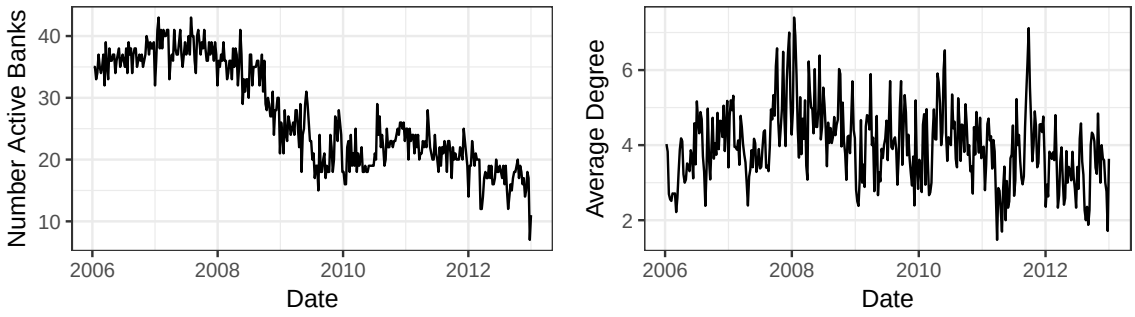


Figure 3.3: Weekly network statistics, including the number of nodes and average degree.

data], but we have seen that the collapse of Lehman Brothers in quarter 39 (2008 Q4) was a major shock for the global financial market in general and the Italian interbank market as well.". Note that as opposed to monitoring in real time, previous works performed historical analyses that utilized ex-post information about when critical events occurred. We utilize these four sub-period definitions to validate our monitoring results.

Similar patterns are also present in network statistics, such as the number of active banks (nodes) and average degree, as shown in Figure 3.3. Note that the network is constructed each week by connecting lender to borrowers, that is, if Bank A lends to Bank B, then an edge is drawn from Bank A to Bank B and weighted by the number of directed loans. We also require that either Bank A or Bank B be one of the publicly-traded banks. Motivating the use of the Hurdle model, we observe a high level of sparsity in the e-MID market network. During the pre-crisis era, there is 73.68% sparsity among all possible bank interactions. This value increases to 77.12% and 83.78% during Crisis 1 and Crisis 2 sub-periods, respectively. In the post-crisis era, sparsity is 76.87%, nearly returning to pre-crisis levels.

Several extant works have focused on characterizing the general network structure within the e-MID interbank lending market. For instance, [39] summarize the degree distribution as heavy-tailed (negative Binomial), but not power-law or scale-free. [38] find that networks from the e-MID consistently exhibit so-called core-periphery structure, where a temporally stable core of banks that comprise 20-30% of the market are actively engaged in

both lending and borrowing. During the financial crisis, the authors find that the reduction of e-MID activity was primarily because of these core banks decreasing their trading. [36] find that the level of temporal aggregation is an important methodological choice, where daily-level network analysis of e-MID interbank lending data looks almost random and uninformative, but meaningful and significant non-random structures appear for longer aggregation periods. As such, in this paper, we present a weekly analysis to improve both interpretability and practical utility for regulators and market participants.

The studies above analyze network statistics, such as density, reciprocity, and clustering coefficient, and find structural breaks around crisis periods. Based on these and related findings, scholars consistently advocate monitoring network statistics that measure particular aspects of connectivity to assess the health and structure of the overall market. This practice is consistent with a majority of previous works in network science that focus on monitoring network statistics to detect temporal changes in the network dynamics [82, 83]. While such an approach is conceptually straightforward, one shortcoming is that the framework does not capture the effect of other covariates, such as country of origin, interest rates, and so on, on the graph formation process. We solve this issue in the proposed methodology, which is presented in detail next.

3.3 Monitoring Sparse Network Sequences with Online Hurdle Models

3.3.1 Overview

We propose a new monitoring methodology for sparse attributed network streams with dynamic structures. The methodology is comprised of modeling the network structure and providing a change detection methodology. In our modeling framework, it is assumed that the edge probabilities are functions of nodes and edges attributes. For example, in the context of financial networks, the probability of a transaction (or the number of transactions) between two banks could be a function of their country of origins, prevailing interest rates, and so on. Although generalized linear models (GLM) have been successfully used

to model attributed networks [6, 40], network sparsity (extreme lack of node connections) violates the assumption to GLMs that the underlying probability distribution should belong to the exponential family of distributions. To address this issue, we use the Hurdle model, which is capable of handling zero-inflated distributions [88]. The hurdle model was used in [50] for modeling network's sparsity; however, in the proposed model the edge and node attributes were not taken into account, and edge probabilities were modeled only as a function time. To take the network dynamics coupled with edges and nodes' attributes into account, we integrate the state-space model with the Hurdle model, where it is assumed that the parameters of the Hurdle regression follow a Markovian process, and develop a sequential estimation scheme using Extended Kalman Filters (EKF) to update the state space parameters and predict the value of upcoming networks. The overall framework is illustrated in Figure 3.4. As shown in the figure, in the offline phase, using a stream of in-control networks, we build a Hurdle model using nodes and edges attributes and estimate the initial state-space parameters. In the online phase, as new network observations arrive, the estimated Hurdle is used to predict the edge values for the incoming network snapshot. Additionally, with the upcoming network observations, the parameters of the state-space Hurdle model are updated using EKF. After calculating the prediction of the new network, the residuals, defined as the difference between observed values and predicted values, are calculated. Residuals can be a proper statistics for network monitoring since they are independent and they exclude the network dynamics. Hence, we can monitor residuals to detect a sudden/structural change in the network. To do this, we fed the residuals into the change-detection module that includes a set of Exponentially Weighted Moving Average (EWMA) control charts. EWMA control chart is chosen for monitoring because of its memory based property.

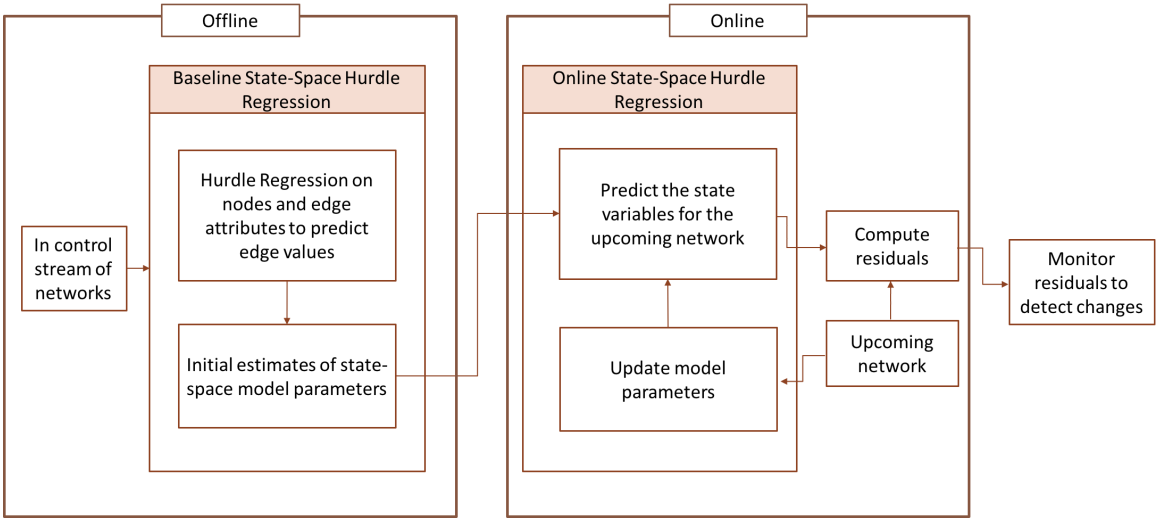


Figure 3.4: Overview of the proposed network monitoring methodology.

3.3.2 Hurdle Models

As mentioned earlier, in attributed networks, the edge probabilities can be defined as functions of nodes and/or edge attributes and often modeled by using GLMs. However, in practice, (financial) networks are often sparse [3, 31], where typically each node interacts with only a few other nodes with which it shares some common attributes and characteristics. Moreover, when a network is studied in a short snapshot (i.e., daily as opposed to weekly), the level of network sparsity increases. When the edge values are in the form of count data, this excessive zero phenomenon is called *zero-inflated* count data.

Regular Poisson models and consequently GLMs, are ineffective in modeling zero-inflated count data. Hence, alternative modeling approaches including zero-inflated [67] and Hurdle [88] models have been proposed. Zero-inflated models assume that the occurrence of zero counts comes from two different origins, *structural* and *sampling*. The zeros of the sampling origins are zeros generated from regular Poisson distribution, which assumes that those zero observations happen by chance. The zeros of the structural origin are because of some specific structure in the data and follow a different distribution. On the other hand, a Hurdle model assumes that all zero occurrences are the results of a structural origin, while the non-zero data have a different origin, following a truncated Poisson

distribution. Therefore, Hurdle models separate the occurrence of zero counts completely from the occurrence of positive counts.

Driven by the observed sparsity in the e-MID interbank lending networks, we utilize the Hurdle model to explain and predict edge formation over time. We choose Hurdle over zero-inflated since we are facing only one type of zero observations. In other words, since in these networks, zero means no connection between two banks, we are considering it as a structural zero and separate its distribution from positive values. The fundamental idea behind the Hurdle model is to generate count variables in a two-stage process. In the first stage, a binary process specifies whether the count value is zero or positive. Given the first stage indicating a non-zero value (i.e., if the hurdle is crossed), another stochastic process generates positive counts. In this Hurdle model, the two-stage process mimics the actual decision that a bank would make about whether to participate in the e-MID market with another bank. If so, the positive count shows the count of transactions between two banks. We would expect the binary process to be heavily weighted towards zero counts for all possible edges.

In Hurdle models, we have two parts in the model. In the first part, we assume that count is equal to zero with probability f_0, and with probability $(1 - f_0)$ count is positive. Given that count is positive, the positive count value, k, comes from the $f_1(k)$ with the associated truncated $f_1(k)/(1 - f_1(0))$ probability (to ensure that zero count does not occur in this case). Note that the positive probability needs to be multiplied by $(1 - f_0)$ to certify that probabilities sum to one [17]. Different hurdle models can be introduced based on the choice of f_0 and f_1. For our Hurdle model, we focus on the "Poisson-logit" specification, where f_0 is a Bernoulli distribution, and f_1 is Poisson distribution. The probabilities are shown in Equation 3.1.

$$P(w_{i,j,t} = k) = \begin{cases} \pi_{i,j,t} & k = 0 \\ (1 - \pi_{i,j,t}) \dfrac{\exp\left(-\lambda_{i,j,t}\right)\lambda_{i,j,t}^{k}}{k!(1 - \exp\left(-\lambda_{i,j,t}\right))} & k \geq 1, \end{cases} \tag{3.1}$$

where $w_{i,j,t}$ is the value of the edge between node i and j at time t, $\pi_{i,j,t}$ is the probability of having no edge between nodes i and j, and $\frac{\exp(-\lambda_{i,j,t})\lambda_{i,j,t}^{k}}{k!(1-\exp(-\lambda_{i,j,t}))}$ is the probability distribution function of a truncated Poisson process at zero (also known as Positive Poisson; [45]). Equation 3.1 can clearly be decomposed as the mixture of a Bernoulli distribution with parameter $\pi_{i,j,t}$ and a Positive Poisson distribution with parameter $\lambda_{i,j,t}$.

To extend this model to a regression setting, we assume that model parameters, i.e., $\lambda_{i,j,t}$ and $\pi_{i,j,t}$, are functions of covariates, i.e., edges and nodes attributes. To write our model, we first define the indicator random variable $d_{i,j,t} = \begin{cases} 0 & w_{i,j,t} = 0 \\ 1 & w_{i,j,t} \geq 0 \end{cases}$ as the indicator of positive occurrence, and $w^{+}_{i,j,t}$ as the positive values of edge weights. For simplicity, we denote $f_0(d_{i,j,t};\pi_{i,j,t})$ and $f_1^{+}(w^{+}_{i,j,t};\lambda_{i,j,t})$ as distributions of zero and positive counts, respectively. The Hurdle regression, using logit and exponential link functions [89] can be written as,

$$\begin{aligned} d_{i,j,t} &\sim f_0(\pi_{i,j,t}) \\ \pi_{i,j,t} &= \text{logit}^{-1}(z_{i,j,t}\beta_{0,t}) \end{aligned} \tag{3.2}$$

and

$$\begin{aligned} w^{+}_{i,j,t} &\sim f_1^{+}(\lambda_{i,j,t}) \\ \lambda_{i,j,t} &= \exp(x_{i,j,t}\beta_{1,t}), \end{aligned} \tag{3.3}$$

where $x_{i,j,t}$, and $z_{i,j,t}$ are covariates (i.e., edges and nodes attributes) used in Bernoulli and Positive Poisson count models, respectively. The covariates used in each model can be same or different. To estimate the model parameters, $\beta_{0,t}$ and $\beta_{1,t}$, the following log-

likelihood should be maximized.

$$\mathscr{L}(\beta_0, \beta_1) = \sum_{\{i,j \in \mathscr{N}} \sum_{,j \neq i\}} \{d_{i,j,t} \times [\log(f_0(w_{i,j,t} = 0; \beta_{0,t} | z_{i,j,t})] + d_{i,j,t} \times [\log(f_0(w^+_{i,j,t}, \beta_{0,t} | z_{i,j,t}) + \log(f^+_1(w^+_{i,j,t}, \beta_{1,t} | w_{i,j,t} > 0, x_{i,j,t}))]\}, \tag{3.4}$$

where $\mathscr{N}$ is the set of total nodes in the network. Assuming independence among $f_0(\beta_0, z)$ and $f^+_1(\beta_1, x)$, the log-likelihood function in Equation 3.4 can be written as a sum of two separate components, namely, $\sum_{\{i,j \in \mathscr{N}} \sum_{,j \neq i\}} \{d_{i,j,t} \times [\log(f_0(w_{i,j,t} = 0; \beta_{0,t} | z_{i,j,t})] + d_{i,j,t} \times [\log(f_0(w^+_{i,j,t}, \beta_{0,t} | z_{i,j,t})\}$ and $\sum_{\{i,j \in \mathscr{N}} \sum_{,j \neq i\}} \{d_{i,j,t} \times [\log(f^+_1(w^+_{i,j,t}, \beta_{1,t} | x_{i,j,t})\}$. Hence, each component can be maximized individually resulting in the following estimates:

$$\begin{aligned} \hat{\pi}_{i,j,t} &= \text{logit}^{-1}(z_{i,j,t} \hat{\beta}_{0,t}) \\ \hat{\lambda}_{i,j,t} &= \exp(x_{i,j,t} \hat{\beta}_{1,t}), \end{aligned} \tag{3.5}$$

where $\hat{\beta}_{0,t}$ and $\hat{\beta}_{1,t}$ are the estimated regression coefficients at time t. Next we discuss how to incorporate network structural dynamics through a state space model on the parameters of the Hurdle model, and use the Extended Kalman Filter (EKF) to estimate and update model parameters over time.

3.3.3 State Space Models and the Extended Kalman Filter

State-space models provide a flexible framework for modeling dynamic systems. In this approach, although the actual state of the system is unknown, it can be inferred over time using noisy observations. In the context of attributed network streams, we assume that the coefficients of the Bernoulli and truncated Poisson regression (β_0 and β_1) are the state variables, which are driven by a stochastic process and the observed edge values, $w_{i,j,t}, (t = 1, 2, ...)$, are noisy observations. Therefore, the state-space Hurdle model is de-

fined by the following equations

$$\begin{aligned} \beta_t &= \mathbf{F}\beta_{t-1} + \epsilon_t \\ w_t &= h(x_{i,j,t}, z_{i,j,t}, \beta_t), \end{aligned} \tag{3.6}$$

where $\beta_t = [\beta_{0,t}, \beta_{1,t}]$ is the state vector, $w_t = \text{vec}[w_{i,j,t}]$ is the vectorized adjacency matrix containing the noisy trades between pairs of nodes, $\mathbf{F}$ is the state transition matrix, and $\epsilon_t \sim N(\mathbf{0}, \mathbf{Q})$ is the process noise with mean $\mathbf{0}$ and covariance matrix $\mathbf{Q}$. h is a non-linear function generating a realization of $w_{i,j,t}$ given the state of the system β_t and the vector of covariates, i.e., $x_{i,j,t}$, and $z_{i,j,t}$. It is assumed that given attributes the interactions between nodes are independent.

In the case of linear state-space models, the Kalman Filter (KF) procedure achieves the optimal estimate of the states [61]. However, as the observation model in Equation 3.6 is nonlinear, we employ the EKF which is shown to be effective in incorporating nonlinearity in parameter estimation [30, 14]. Similar to KF, EKF provides a recursive estimation procedure that only uses the current network snapshot (at time t) and the previous parameter estimates (at time $t-1$) to update the parameter estimates. EKF uses the Taylor expansion to linearize the nonlinear observation function, $h(x_{i,j,t}, z_{i,j,t}, \beta_t)$, and then applies the KF estimation equations. Specifically, given $\mathbf{F}$ and $\mathbf{Q}$, the EKF for the state-space Hurdle regression can be summarized as follows.

Prediction Step

Let $\beta_{t|t-1}$ and $\mathbf{P}_{t|t-1}$ denote the Kalman predictions of the state β_t and its covariance matrix given observation until time $t-1$ (w_l; $l = 1, ..., t-1$,), and let $\beta_{t|t}$ and $\mathbf{P}_{t|t}$ denote the estimation of the state and its covariance matrix, given observations until time t (w_l; $l = 1, ..., t$). Now using the previous estimates, the prediction equations at time t are given by

$$\begin{aligned}
\beta_{t|t-1} &= \mathbf{F}\beta_{t-1|t-1} \\
\mathbf{P}_{t|t-1} &= F\mathbf{P}_{t-1|t-1}F^T + \mathbf{Q}, \quad t = 1, 2, \dots
\end{aligned} \tag{3.7}$$

where the initial estimates, i.e., $\beta_{0|0}$ and $\mathbf{P}_{0|0}$ can be obtained from fitting a Hurdle model to the first network snapshot data.

Update Step

At time t, incoming network data, i.e., w_t, are used to update the predicted parameters using the set of equations,

$$\begin{aligned}
\mathbf{K}_t &= \mathbf{P}_{t|t-1}{\mathbf{H}_t}^T(\mathbf{H}_t\mathbf{P}_{t|t-1}{\mathbf{H}_t}^T + \mathbf{R}_t)^{-1} \\
\beta_{t|t} &= \beta_{t|t-1} + \mathbf{K}_t(w_t - h(x_t, z_t, \beta_{t|t-1})) \\
\mathbf{P}_{t|t} &= (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\mathbf{P}_{t|t-1},
\end{aligned} \tag{3.8}$$

where $\mathbf{H}_t = [\frac{dh}{d\beta}]_{\beta=\beta_{t|t}}$ is the measurement Jacobian matrix used for linearization of the observation function $h(x_{i,j,t}, z_{i,j,t}, \beta_t)$, $\mathbf{K}_t$ is known as Kalman gain, and $\mathbf{R}_t$ is a covariance matrix of observations at time t, which depends on the distribution of observations. Specifically, for Bernoulli observations $R_{i,j,t} = (1 - \hat{\pi}_{i,j,t})\hat{\pi}_{i,j,t}$ and for Positive Poisson $R_{i,j,t} = \frac{\hat{\lambda}_{i,j,t}}{1 - \exp(-\hat{\lambda}_{i,j,t})}(1 + \hat{\lambda}_{i,j,t} + \frac{\hat{\lambda}_{i,j,t}}{1 - \exp(-\hat{\lambda}_{i,j,t})})$.

The detailed derivation of the prediction and update equations for EKF can be found in [14]. In practice, the state transition matrix $\mathbf{F}$ and the state covariance matrix $\mathbf{Q}$ are unknown. In the Appendix C, we provide the estimation procedure for these matrices using a series of in-control network snapshots.

3.3.4 Monitoring Approach

In this section we propose a monitoring procedure in order to detect structural changes in sparse attributed networks. Since we are mainly interested in changes caused by a shock factor rather than the dynamic changes in the model, we propose to monitor the residuals computed from the EKF applied on the Hurdle Regression model. To do this, we fit the data incrementally using our model. The in-control set of networks are denoted as $\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_T$. For each network snapshot $\mathbf{W}_i$, network parameters are predicted using the EKF and Hurdle model. Then, the vector of updated parameters $\beta_{t+1|t}$ is used to predict the adjacency matrix at time t for time $t+1$ using the appropriate link functions discussed above. Hence $\hat{w}_{i,j,t+1} = h(x_{i,j,t}, z_{i,j,t}, \beta_{t+1|t})$ using Equation 3.6. Moreover, at each time and for each edge between nodes i and j, we define residual as the difference between observed value and predicted value of an edge value. Therefore, $\hat{\epsilon}_{i,j,t} = w_{i,j,t} - \hat{w}_{i,j,t}$. Since the models capture the dynamic in the data, the residuals exclude the ordinary network dynamics. Moreover, residuals are approximately independent over time. Hence, they can be a proper statistic to monitor the network and detect the structural changes. Since residuals are obtained from non-normal models, the observation variance is heteroscedastic, i.e., the variance is not necessarily constant over time. To have approximately constant variance, we use Pearson residuals, denoted by $r_{i,j,t}$, which is computed as,

$$r_{i,j,t} = \frac{\hat{\epsilon}_{i,j,t}}{\sqrt{var(\hat{w}_{i,j,t})}},$$

where $var(\hat{w}_{i,j,t})$ is the estimated variance of the observation, and can be calculated knowing the predicted observation and its probability distribution (Positive Poisson or Bernoulli). If the process is in-control, Pearson residuals asymptotically follow an independent standard normal distribution. Hence, we can use statistical process control charts to monitor the residuals and detect the changes. We chose the Exponentially Weighted Moving Average (EWMA) control chart for monitoring, which is a control chart used to monitor small

shifts in the process by incorporating memories of the previous observations in calculating the monitoring statistic. The EWMA weights observations in a geometrically declining order such that the newest observations have higher weights while the oldest ones have much smaller weights. At each time, we have a total of m residuals calculated where m is the total number of edges in a network. To calculate one collective statistic, we define $\bar{r}_t = \frac{\sum_{i,j} r_{i,j,t}}{m}$ and monitor $\bar{r}_t$ over time. The EWMA statistic corresponding to $\bar{r}_t$ is denoted by ω_t and calculated as

$$\begin{aligned} \omega_t &= \lambda \bar{r}_t + (1-\lambda)\omega_{t-1} \quad t \geq 1 \\ \omega_0 &= 0, \end{aligned} \tag{3.9}$$

where $\lambda \in [0, 1]$ is a constant that specifies the depth of memory. Higher lambda gives more weights to the current observations, and smaller lambda gives more weights to the previous observations. The control limits are defined as

$$\begin{aligned} UCL &= \mu_0 + k\sigma_0\sqrt{\frac{\lambda}{2-\lambda}} \\ LCL &= \mu_0 - k\sigma_0\sqrt{\frac{\lambda}{2-\lambda}}, \end{aligned} \tag{3.10}$$

where μ_0 and σ_0 are the mean and standard deviation of offline (training) Pearson errors, and k is a parameter that controls the width of control limits. If $\omega_t > UCL$ or $\omega_t < LCL$, we reject the null hypothesis, indicating a change has occurred in the network stream. To obtain parameters k and λ, we use Monte Carlo simulations and aim to find a set of parameters that result in a desired in-control average run length (ARL_0). For this purpose, we generate in-control network streams and monitor them using the aforementioned monitoring method until an out of control alarm (false alarm) is raised. We record the run length (RL) values corresponding to each network stream. We repeat this procedure for 1000 times and calculate the in control average run length (ARL_0) over all iterations. The values

of k and λ should be set so that the ARL remains at a reasonably large value.

3.4 Results from Synthetic Data

In this section, we evaluate the performance of our monitoring methodology against three benchmark methods by using simulations. The first two benchmark methods monitor network statistics using EWMA control charts and are motivated by the approaches advocated the financial economics literature. Specifically, the first benchmark monitors the average degree of the network and the second benchmark monitors the average of network betweenness. These statistics have been used commonly for network monitoring ([82, 83, 48]).The third benchmark utilizes the closest extant Statistical method to our knowledge, the dynamic monitoring approach proposed by [40], which utilizes a dynamic Poisson distribution to model the count data without accounting explicitly for excessive sparsity.

Each simulated network is composed of fifty nodes, hence there are $50\times(50-1) = 2450$ potential directed edges in each network. To match our real data setting, we assume that the number of interactions between nodes i and j is a function of five attributes in the model denoted as $\mathbf{x}_{i,j,t} = \mathbf{z}_{i,j,t} = [x^{(1)}_{i,j,t}, x^{(2)}_{i,j,t}, \ldots, x^{(5)}_{i,j,t}]^T$. The attribute values vary for each edge and time, and are generated using normal distribution with mean $\mu = [0.5, 0.5, 0.5, 0.5, 0.5]^T$ and variance $\Sigma = 0.25 \times \mathbf{I}_{5\times5}$. The relationship between the attributes and the response value follows a dynamic Hurdle model. Therefore, we assume that the binary outcomes (whether there is a connection between two nodes) have a Bernoulli distribution, with probability $\pi_{i,j,t} = \text{logit}^{-1}(x_{i,j,t}\beta_{0,t})$, and the positive edge weight outcomes follow a Positive Poisson distribution with $\lambda_{i,j,t} = \exp(x_{i,j,t}\beta_{1,t})$. Here $\beta_{0,t} = [\beta^0_{0,t}, \beta^1_{0,t}, \beta^2_{0,t}, \ldots, \beta^5_{0,t}]^T$ are the coefficients of the binary model at time t, where $\beta^0_{0,t}$ is the coefficient corresponding to the intercept. Similarly, $\beta_{1,t} = [\beta^0_{1,t}, \beta^1_{1,t}, \beta^2_{1,t}, \ldots, \beta^5_{1,t}]^T$ are the coefficients of the Positive Poisson model at time t.

To simulate a dynamic stream of networks, we assume the underlying state transition model with $\beta_{0,t} = \mathbf{F}\beta_{0,t-1} + \epsilon_{0,t}$ and $\beta_{1,t} = \mathbf{F}\beta_{1,t-1} + \epsilon_{1,t}$. Here we set $\epsilon_{0,t} \sim \mathbb{N}(0, \mathbf{Q})$

and $\epsilon_{1,t} \sim \mathbb{N}(0, \mathbf{Q})$. In the simulations, we use $\beta_{0,0} = [0.01, 0.01, 0.01, 0.01, 0.01, 0.01]$ and $\beta_{1,0} = [0.2, 0.2, 0.2, 0.2, 0.2, 0.2]$, $\mathbf{F} = 0.8\mathbf{I}_{6\times 6}$, and $\mathbf{Q} = 0.25\mathbf{I}_{6\times 6}$. We use in-control simulated snapshots of networks to estimate the control chart and calculate the EWMA control limits based on methods described in Section 3.3.

We evaluate the performance of our method by simulating three scenarios, each of which induces changes to specific coefficients underlying the network process to create out of control situations. For each selected coefficient β_i, the shift is $\delta\sigma_i$, where δ is a constant representing the magnitude of the shift and σ_i is the standard deviation of i^{th} coefficient for in-control situation, which is equal to $\sigma_i = \sqrt{\frac{Q_{ii}}{(1-F_{ii})^2}} = 2.5$. Therefore, for the Bernoulli model the changed coefficient will be $\beta^i_{0,\tau} = F_{ii}\beta^i_{0,\tau-1} + \epsilon^i_{0,\tau} + \delta\sigma_i$ and for Positive Poisson model the changed coefficient will be $\beta^i_{1,\tau} = F_{ii}\beta^i_{1,\tau-1} + \epsilon^i_{1,\tau} + \delta\sigma_i$ at time τ, i.e., the coefficients are shifted by $\delta\sigma_i$ at time τ.

Scenario I represents a case where the change point is affecting the underlying dynamics in two ways. First, it affects the decision of whether two nodes are interacting. Secondly, it affects the level of interaction (weights) after the first decision is made. In other words, we assume the change has affected both Bernoulli and Positive Poisson model. In each model, we apply the change in three out of six coefficients. So, we assume that coefficients $\beta^2_{0,\tau}$, $\beta^4_{0,\tau}$, $\beta^5_{0,\tau}$ of Bernoulli model, and $\beta^2_{1,\tau}$, $\beta^4_{1,\tau}$, $\beta^5_{1,\tau}$ of Positive Poisson model change at time τ.

Scenario II represents a case where the change only affects the decision of whether two nodes interact. However, after this decision is made, the Positive Poisson model coefficients remain unchanged. Thus, we assume that change has only affected the coefficients of Bernoulli model, where $\beta^2_{0,\tau}$, $\beta^4_{0,\tau}$, $\beta^5_{0,\tau}$ from the Bernoulli model have changed at time τ.

Scenario III represents a case where the change only affects the amount of interaction (edge weight) between two nodes. The coefficients determining whether two nodes interact remains unchanged, but $\beta^2_{1,\tau}$, $\beta^4_{1,\tau}$, $\beta^5_{1,\tau}$ from the Positive Poisson model have changed at time τ.

Here we briefly explain how we perform each benchmark monitoring approaches. To monitor the average degree and average network betweenness, in Phase I analysis (in-control analysis), we simulate in-control networks with the characteristics explained above. Afterward, we calculate network degree and betweenness values and compute the average over all observations at each time. Then, we use the average statistic to build the EWMA statistic as explained in Equation 3.9. We choose λ and k such that in-control ARL (ARL_0) is equal to 200. For each out-of-control scenario, we generate out-of-control networks based on scenario explained and monitor network statistic based on the control limits specified in Phase I. Modeling with Dynamic GLM approach is very similar to our work. In this case, instead of assuming that network comes from a Dynamic two-part Hurdle Model, we assume that edge counts are from a Dynamic one part Poisson Model. Hence, we assume that edge counts have a Poisson Distribution at each time, while the mean of the distribution λ is a function of network attributes, $x_{i,j,t}$, i.e. $\lambda \sim exp(x_{i,j,t}, \beta_P)$. Using EKF and Poisson regression, and employing the networks attributes, we fit the model over time and calculate the residuals. Moreover, at each time we calculate the Pearson residuals and monitor it using EWMA statistic. Similarly, we determine the parameters and control limits such that ARL_0 is equal to 200. For each scenario, we monitor the statistic with the control limits specified in Phase I.

To evaluate the performance of each method we use the Average Run Length (ARL), which measures how quickly the method detects the change induced in each scenario for different values of δ. Specifically, we simulate networks for an out of control scenario until an out of control alarm is raised. Each time the alarm is raised, we record the run length (RL), which is the number of simulated networks (time points) until the change is detected. We iterate this procedure 1,000 times and record the Average Run Length (ARL) over all iterations so that a method with smallest ARL for an out of control situation represents superior ability in detecting the change. Also note that to ensure that all methods can be compared relatively using the out of control scenarios, we specify control limits such that

the in-control ARL (ARL_0) for all methods is equal to 200 ($\alpha = 0.005$) and subsequently use these tuned control limits for detection in the out of control scenarios.

The ARL results are shown in Figure 3.5 for all scenarios. As can be readily observed, for all scenarios, monitoring network statistics (degree and betweenness) has the highest ARL (worst performance), showing that static methods should not be preferred for the explicit purpose of detecting changes in networks. We also see from the figure that all methods perform better in the first scenario, where the coefficients are shifted in both the Bernoulli and Positive Poisson model. Yet, in this scenario, we can see that for a change as small as $\delta = 0.75$ our method has $ARL \approx 2$, while Dynamic GLM method has $ARL \approx 35$, and monitoring network statistics methods have $ARL \approx 200$. Hence, for this small shift, our method almost instantly detects the change while it takes on average 35 runs for Dynamic GLM method to detect this shift; the other methods are not capable of detecting this small shift.

For Scenario II, all methods have higher ARL in comparison to other scenarios, which is due to the fact that shift is imposed only on the existence of an edge while the edges weights remained intact, hence detecting such shift is more challenging. In this case, approaches based on monitoring network statistics are not able to detect changes with even large δ shifts. Furthermore, we again see that the dynamic GLM method has significantly higher ARL in comparison to our proposed method. For example, for a shift with $\delta = 3.5$, our method has $ARL \approx 1.5$, while Dynamic GLM method has $ARL \approx 148$, and monitoring network statistics methods have $ARL \approx 200$.

For Scenario III, all methods have slightly higher ARL in comparison to the first scenario, which is due to the fact that shift is imposed only on the Positive Poisson model (the edges weights) while the Bernoulli model (decision to connect) remains intact. We can again observe that for a change as small as $\delta = 1$ our method almost instantly detect the change ($ARL \approx 1$), while Dynamic GLM method has $ARL \approx 22$ and monitoring network statistic methods have $ARL \approx 200$.

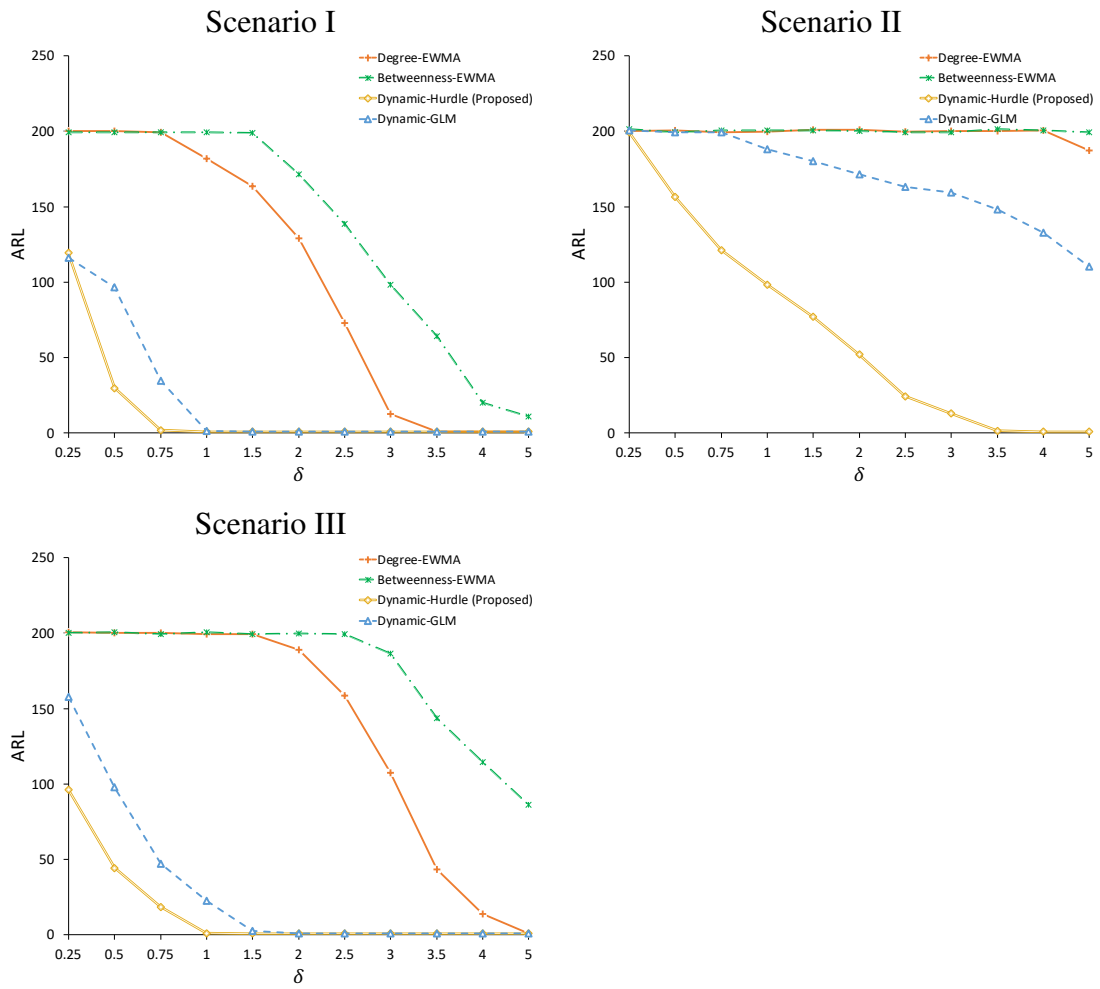


Figure 3.5: Average Run Length comparison of methods based on simulated data for different magnitude of shifts (δ).

Thus, by comparing our proposed method with other methods in all three scenarios, we observe that our proposed method has the lowest ARL for different magnitude of shifts in all the scenarios. Foreshadowing results with the real data, we see evidence that monitoring network statistics is not well suited for change point detection. We also see that our proposed approach outperforms the dynamic GLM model proposed by [40] in all cases. The gap between our proposed method and the benchmark methods is particularly pronounced in Scenario II, emphasizing the importance of modeling the data's sparsity appropriately.

3.5 Results from e-MID Data

3.5.1 Preprocessing and Model Specification

Having established self-consistency and proper performance of the proposed modeling and estimation framework through simulation, we now turn to demonstrate the method's real-world viability and applicability by using the e-MID data that was described earlier. We begin by reviewing the potential change points due to events relating to the financial crisis, discussing the model specification and training, followed by a detailed discussion of the findings.

Using events reported in previous works [15, 36], as shown in Figure 3.1, we have potentially three major change points creating four sub-periods in the data: (1) a pre-crisis period from January 2, 2006, until August 7, 2007, when the European Central Bank (ECB) noted worldwide liquidity shortages; (2) the first crisis period from August 8, 2007 until September 12, 2008, when Lehman Brother's collapsed; (3) the second crisis period from September 16, 2008, through April 1, 2009 when the ECB announced the end of the recession; and (4) the post-recession period, from April 2, 2009, onwards.

We use a number of nodes (bank) and edges attributes listed in Table 3.1 as independent variables in the Hurdle model, including whether the two banks are originating from different countries, interest rates, returns correlation, and so on. Country Difference is motivated by [36], who found that Italian banks tended to trade with other Italian banks a vast majority of the time. Thus, we expect this variable to be significant, especially for modeling whether two banks have any trades with each other. Most of the other variables are based fundamentally on stock market returns. As shown theoretically in [15] and references therein, bank activity in the interbank market can be influenced by their stock market performance, especially when the impact on the bank's balance sheet is substantial. As such, we expect returns-based variables to be important particularly when the stock market is volatile, i.e., in crisis sub-periods.

Table 3.1: Independent variables used in the Hurdle model. The variables Amount and Rate are used only in the Positive Poisson regression when conditioning on the existence of an edge.

Node attributes

Variable	**Description**
Lender's return	The average stock market return of the lending bank in the current week
Borrower's return	The average stock market return of the borrowing bank in the current week

Edge attributes

Variable	**Description**
Return Correlation	The correlation between the two banks' returns from the start of the data up to the current week
Amount	The average number of transactions between two banks in the current week (if any transaction occurred)
Rate	The average interest rate of each loan between two banks in the current week (if any transaction occurred)
Country Difference	An indicator variable that is one if two banks are from different countries

Using the notation given above, where $\mathbf{X}_t$ are independent variables to model whether two banks have any trade and $\mathbf{Z}_t$ are independent variables for the Positive Poisson model, we assume $\mathbf{X}_t = [x_{i,j,t}]$ includes all attributes in Table 3.1, and $\mathbf{Z}_t = [z_{i,j,t}]$ includes all attributes except for Amount and Rate. Note that we are not including Amount and Rate in the Bernoulli model because this information is generated only after a transaction occurs.

We use a hierarchical analysis approach to simulate how the methodology would have performed if implemented in real time. Specifically, the first 20 weeks of pre-crisis data are used as offline observations to obtain appropriate values of F, Q as well as initial estimates of the regression coefficients. Also based on the Pearson residual errors from the offline data, we determine the control limits for ARL_0 (in-control ARL) to be equal to 200. Starting with week 21, we enter the online monitoring phase (see Figure 3.4 to review the methodology). Once a change point is detected, the methodology is restarted, with retraining of the model using the next 20 weeks as offline data before entering another monitoring phase.

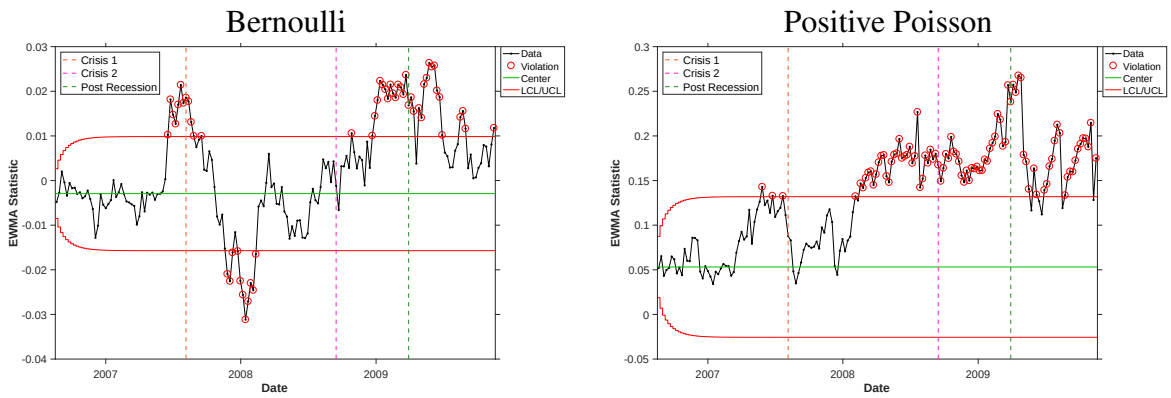


Figure 3.6: EWMA charts for Pearson Residuals from the proposed model to detect the onset of Crisis 1.

3.5.2 Results

Figure 3.6 shows the EWMA control charts for Logistic and Positive Poisson regressions, where we can see that both aspects of the Hurdle model "raise the alarm" before the official announcement that marks the beginning of the first crisis sub-period. Specifically, the Positive Poisson model would have alerted regulators the week starting on May 28, 2007, and the Bernoulli model would have alerted regulators the week of June 18, 2007.

To compare the performance of our method with financial economics monitoring approaches, we provide the EWMA control charts for monitoring network's average degree and betweenness in Figure 3.7. As the results show, these control charts fail to raise an alarm before the official announcement of the first crisis sub-period, providing evidence that network statistics may serve as a weak foundation for an early-warning system.

When monitoring using our approach, since the EWMA statistic is representing the difference between the actual and estimated value, observing its trend during Post-Crisis can help us interpret the changes in comparison to the Pre-Crisis era. For the Bernoulli model, the EWMA statistic is sharply negative at the onset of the first crisis sub-period, which means that the model over-estimates existence of edges (loans between banks). In other words, for two banks, the probability that they have any transaction sharply decreases at the start of the crisis. Adding evidence that activity in the interbank market dropped

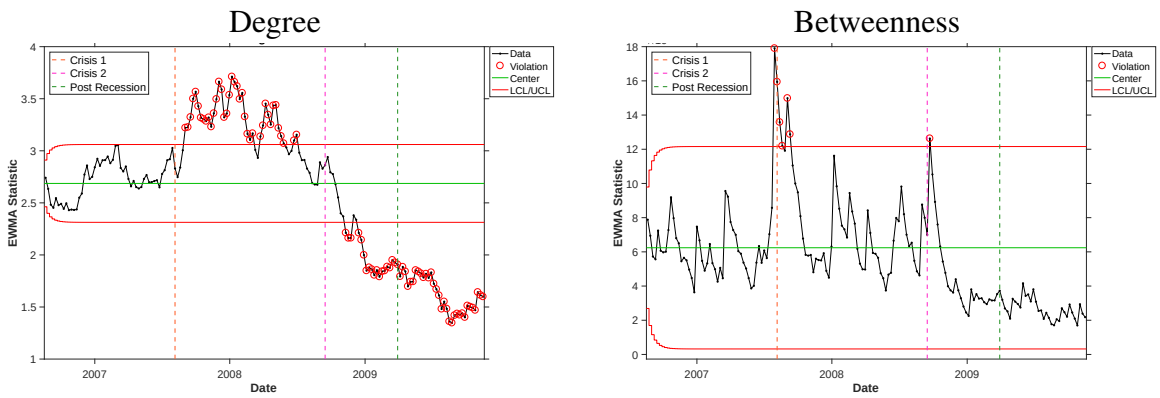


Figure 3.7: EWMA charts for networks statistics (degree, and betweenness) to detect the onset of Crisis 1.

precipitously, as shown in Figure 3.8, the regression coefficient for Country Difference is consistently negative indicating that banks generally prefer to trade with other banks based in the same country. The U shape shows that as the crisis unfolded, banks became even less willing to have transactions with the banks from other countries, but as the crisis concluded and in the Post-Recession sub-period trading activity (specifically counter-party trust) was returning to normal. Similarly, for the Positive Poisson model, we can see, in contrast to activity during the crisis, an apparent increasing trend within the Post-Recession sub-period with coefficients ending close to zero. Thus, by the end of 2012, the number of transactions among two connected banks is not affected by country differences. The estimated coefficients for the Amount and Rate variables in Figure 3.9 also show meaningful patterns. There is a clear increasing trend in Amount denoting that post-recession, banks were able to obtain more funding in comparison to before this era. The coefficient for Rate was positive during the crisis, but negative in the post-recessionary period. One potential explanation is that banks that wanted funding had to pay higher rates during the crisis (i.e., it was a lender's market), but after the crisis, interbank funding was more readily available. Estimated coefficients for other independent variables are not shown since they were centered on zero without meaningful trends. Overall, in addition to detecting the onset of the crisis in real-time, the detailed results are consistent with accepted narratives about the crisis,

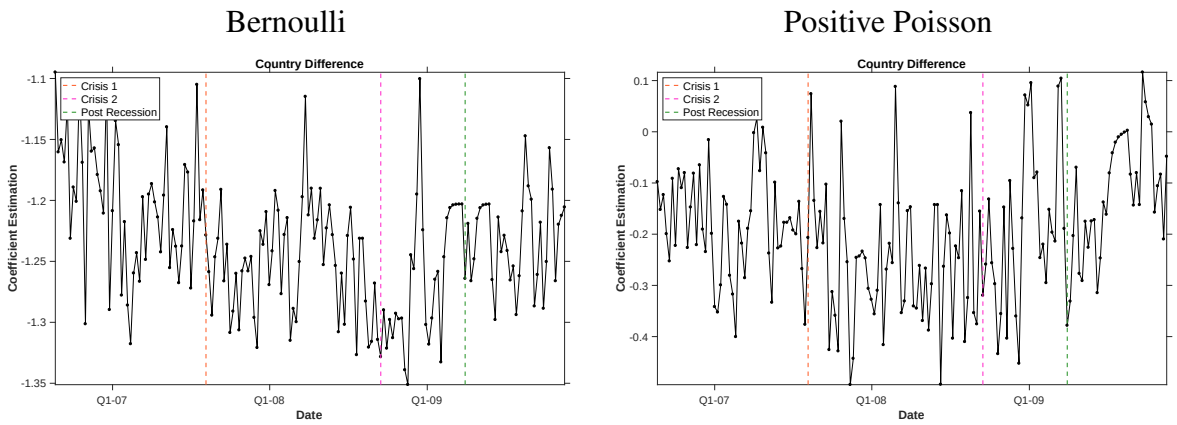


Figure 3.8: Estimated Coefficients for Country Difference in the Bernoulli and Positive Poisson models starting from Pre-Crisis data.

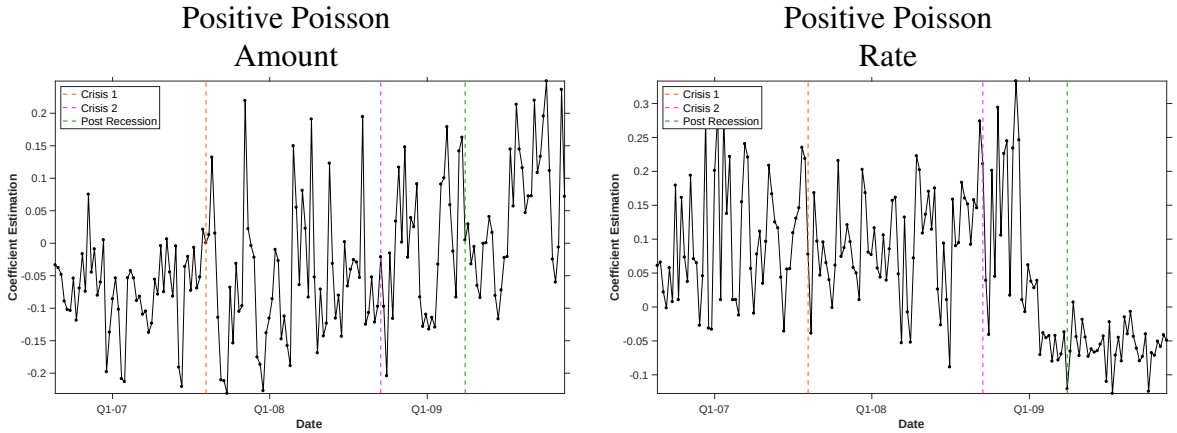


Figure 3.9: Estimated Coefficients for Amount and Rate in the Positive Poisson model starting from Pre-Crisis data.

where trust in the interbank market dropped markedly causing banks to stop participating entirely in the e-MID market, followed by a return to pre-crisis conditions [15, 36].

In practice, it can be important to identify possible root causes for the alarms, which can be accomplished by estimating the actual date of the change point. [90] proposed a method to estimate the EWMA change point after receiving an out of control signal at time T. In this method if the out of control signal is raised when monitoring statistic is above the upper control limit (UCL), the estimated change point is $\hat{\tau} = \max[i : z_i \leq \mu_0, i \leq T]$, i.e., the estimated change point $\hat{\tau}$ is the first point before the alarm time when the EWMA statistic is below the center line μ_0. Similarly, if the out of control signal is raised when

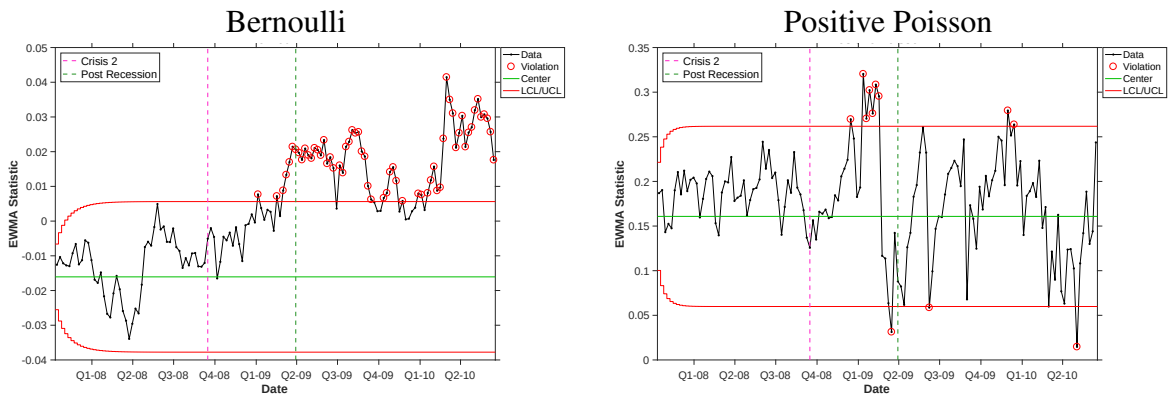


Figure 3.10: EWMA charts for Pearson Residuals from the proposed model to detect the end of the financial crisis and start of the post-recessionary period.

monitoring statistic is below lower control limit (LCL), the estimated change point is $\hat{\tau} = \max[i : z_i \geq \mu_0, i \leq T]$, i.e., the estimated change point $\hat{\tau}$ is the first point before the alarm time when the EWMA statistic is above the center line μ_0. Using this heuristic, we find a change to crisis conditions in the Bernoulli model dated to the week of May 28, 2007, and for the Positive Poisson model dated to the week of March 12-16, 2007.

Moving to the detection of the onset of Crisis 2 and the Post-Crisis eras, we use the first 20 weeks of Crisis 1 era for offline training. The EWMA control charts for Bernoulli and Positive Poisson regression are shown in Figure 3.10, where we see that both parts of the Hurdle model do not detect any change in the interbank market around the failure of Lehman Brothers (the onset of Crisis 2). Both models do successfully capture the change before the onset of Post Recession announcement. The Bernoulli model raises the alarm the week of January 5, 2009, with the change point dated to October 6-10, 2008. The Positive Poisson model raises the alarm the week of December 15, 2008, with its change point dated back to November 3, 2008.

Inspecting the trend in the EWMA statistics, we see increases for the Bernoulli and Positive Poisson models before the onset of the Post-Recession period that continue to the end of the data, demonstrating that banks returned to the market as overall conditions and trust levels improved. A very similar pattern emerges as previously reported when inspecting the estimated coefficients for independent variables (figures shown in the Appendix).

3.5.3 Validation and Discussion

Here we discuss the reported dates corresponding to the alarms and change points.

We first note that all alarms under the proposed model would have been raised prior to official announcements by the ECB, underscoring that the proposed model could be a valuable tool for monitoring financial markets. While we do not know precisely when governments realized internally there was a crisis, we find evidence that the model results were ahead of official policy. For instance, both aspects of the Hurdle model raised alarms by June 18, 2007. On June 6, the ECB raised interest rates, followed by the publication of the Financial Stability Review [28] on June 15, which struck a cautiously optimistic tone. The Financial Stability Review included positive outlooks, with statements such as "Looking forward, with the euro area financial system in a generally healthy condition and the economic outlook remaining favorable, the most likely prospect is that financial system stability will be maintained in the period ahead." Such forecasts were coupled with warnings about how the financial system was growing particularly vulnerable "to an abrupt and unexpected sharp decline in market liquidity", which underscores the importance of our results given that we are studying an interbank market – a key source of liquidity for banks.

Similarly, the alarms signifying the end of the crisis using the proposed methodology is raised between December 15, 2008, to January 5, 2009. Coincidentally, the ECB published another Financial Stability Review on December 15 [29] stating that "The extraordinary remedial actions taken by central banks and governments, which are aimed at addressing liquidity stresses and strengthening capital positions, thus contributing to restoring confidence in, and improving, the resilience of financial systems, were successful in stabilizing the euro area banking system." This period also coincides with an intense activity by the U.S. Treasury Department as a result of a new law giving it broad new powers (see discussion of the Emergency Economic Stabilization Act below) to strengthen the financial and auto sectors of the U.S. economy [33].

To further validate our results, we consider whether the identified change point dates match previously reported results or known events. The earliest detected change point for Crisis 1, the week of March 12, 2007, for the Positive Poisson regression, closely follows the root-cause event as identified by the [33] of a Freddie Mac announcement on February 27, 2007, that they would no longer buy the riskiest type of mortgages (sub-prime). Similarly, the earliest identified change point signaling the end of the crisis was October 6-10, 2008 for the Bernoulli model, which coincides with the so-called bank bailouts (the Emergency Economic Stabilization Act of 2008; [114]) being signed into law by then President Bush on October 3, 2008.

3.6 Conclusion

Developing a rigorous monitoring system that helps in identifying systemic risk sources and vulnerabilities within the financial system, as well as understanding interactions between financial entities is an essential task in itself and also a fundamental priority for regulators with public welfare implications. Ideally, the monitoring system should be applicable to any market or exchange where high resolution, audit-trail trading data is generated. The monitoring system should also be able to incorporate external information about the market participants and observed transactions, and also model appropriately sparse networks since often market participants trade with a relatively small percentage of others. In this paper, we present one novel solution to this problem with the desired properties. The approach combines Hurdle models, a classical approach to zero-inflated count data, with state-space models and Exponentially Weighted Moving Average control charts.

We demonstrated how the model could be applied for monitoring the e-MID interbank lending market, where we found several promising results showing that the proposed model would have raised alarms to regulators prior to official announcements by the ECB. The identified change point dates are also highly interpretable, matching closely with several critical real-world events.

These findings have several implications for exchanges, such as the e-MID, in addition to banks and regulators. From e-MID's perspective, we have shown how their proprietary data that captures the network of interactions can be transformed into an indicator of market changes. As such, the e-MID could explore a new monetization strategy by publishing statistics computed through our proposed approach that regulators and risk managers could purchase in order to understand the health of the interbank market. This idea has promise given that this new data service would not betray confidential information on any specific market participant and still reveal essential dynamics about the overall market. Moreover, banks typically do not have knowledge on the interbank operations of other banks, meaning that they must rely on the e-MID or regulatory bodies to estimate the model and release (or sell) the pertinent statistics. Regulatory entities, like central banks, may have direct access to the relevant network data and could utilize our methodology as a component of a more extensive monitoring system.

Moreover, we showed the effectiveness of our method using simulation studies under different scenarios. The simulation results show that specifically when the change affects the decision whether two nodes interact with each other, the proposed approach is significantly outperforming other methods in the literature. This is because of the fact that the sparsity and zero counts are generated from a different distribution in our model, hence if change affects the network sparsity, our proposed model is very sensitive in detecting it.

There are also several areas of future work both from applications and methodological point of view. In this paper, we studied interbank lending networks. Similar networks can be constructed from other types of markets [11, 1] or even inferred statistically from stock market data [13, 26]; applying the proposed methodology to different markets will require changing the node and edge attributes as well as potentially the Hurdle model itself since the networks may exhibit different or additional properties to sparsity. Further, when the market participant ID is known (in our case it was hidden to preserve confidentiality), it would be useful in practice to monitor several networks (markets) simultaneously since

it becomes possible to link activity across markets. Simultaneous monitoring of multiple networks creates several challenges, from visualization of the data and results to detecting changes in multivariate distributions. As such, our work represents the first rigorous application of monitoring techniques beyond tracking network statistics to financial networks, and the results demonstrate the proposed approach could be a valuable starting point to utilize and extend when monitoring the financial system.

CHAPTER 4
DISCRIMINATIVE DBM FOR CLASSIFICATION AND ITS EXTENSION TO MULTIMODAL INPUTS

4.1 Introduction and Literature Review

Multimedia data consists of one or more media data types, including text, images, speech, audio, and video. With the advancements in storing technology and data collection, MD is available in a wide variety of industries such as manufacturing, online advertising, websites, healthcare, and so on. One example of multimedia data is advertising data that can be available in the form of video, audio, images, text, scripts, etc. In advertising industries, it is imperative to have the proper methods of modeling the effectiveness of ads. For instance, when having a new advertisement, we wish to predict how effective it will be in the marketplace before using it. Moreover, proper insights about multimedia data models can help in creating and designing future ads. Therefore, having a multimedia model that can capture the data structure and relations and predict data labels is imperative. When working with such data, its unstructured, complicated nature hinders general Machine Learning (ML) approaches from extracting meaningful features and modeling the relations. Hence, more advanced methodologies that capture this complicated nature are required. One popular approach used in modeling multimedia data is the Restricted Boltzmann Machine (RBM). RBM is a probabilistic model with one layer of visible units and one layer of hidden units where there is an undirected connection between two layers (but not within them) [37]. The binary units can be used as a more clear representation of the visible units; hence, RBMs can be used for feature engineering and dimension reduction [54]. One important characteristic of RBMs is that, given that there are enough hidden units in the model, an RBM can represent any distribution over binary vectors [37, 69]. Consequently, RBMs are

applied extensively as generative models for various types of data such as images [111, 66], speech [58, 76], text documents [24], etc. RBMs and many existing machine learning methods have "shallow" architecture, meaning they usually have only one layer of hidden units. However, the literature shows that such "shallow" systems would not be able to extract complex structures from complicated, high-dimensional data [10]. Moreover, training such systems requires a significant amount of labeled data. On the other hand, using architectures with multiple nonlinear layers requires much fewer labeled data [72]. Therefore, multilayer generative models were introduced to better model complex data. RBMs were extended to two types of deep networks: Deep Belief Networks (DBNs) [54] and Deep Boltzmann Machines (DBM) [98]. Studies show that these deep networks outperform traditional ML approaches like Support Vector Machines (SVMs) and traditional feedforward neural networks for high-dimensional, complex data [62]. DBN and DBM have the same topological structure; however, DBM has undirected connections among all layers, while DBN has a directed, top-down relationship toward the visible layer. Hence, in DBM's architecture, all layers are representative of the previous layer, while this is not the case in DBN. Therefore, DBM can be seen as a type of Markov random field with undirected connections among layers, making each layer a distinct representation of the input observations. Moreover, in spite of DBN and deep convolutional neural networks, DBM's inference procedure includes a top-down feedback and bottom-up pass. Hence, it can better incorporate the uncertainty of the input data. RBMs and DBMs are generally used as feature extractors for complex data, but these methods cannot be directly applied to classification problems. In order to use the methods for classification, either the extracted features from the model are fed into a discriminative method [42] or the weights learned from the generative models are used for initializing the weights in a classification problem [52]. In [68], authors presented a new training approach for using RBMs directly in classification problems known as ClassRBM. ClassRBM is a probabilities model that incorporates the label data as well as inputs in its architecture. The model tries to capture the

conditional distribution of labels given inputs; hence, ClassRBM aims to find the relationship among inputs and labels. ClassRBM is used frequently in the literature for a variety of classification problems, such as author profiling using emails and blogs [5], radar target recognition [92], breast cancer prediction, classification problems in medical domains [113, 112], credit risk classification [74], and so on.

Due to estimation intractability, an effective deep extension of ClassRBM has not been used in the literature. In this paper, we propose a new estimation approach for deep ClassRBM (ClassDBM) when we have more than one hidden layer. For training the network, inspired from the work in the literature, we present an algorithm using Contrastive Divergence (CS) and Mean Field (MF) Approximation to learn the model weights. Moreover, to predict the new observations using the learned model, we propose an algorithm that selects the label with the highest Mean Field estimation probability. Finally, we evaluate the performance of our proposed methodology by implementing it on two benchmark machine learning image data. In this part, we compare our method with ClassRBM and traditional deep learning methods. We also implement our approach on real-world advertising data to predict the effectiveness of the ads and present new insights on how to create better quality ads.

The rest of the paper is arranged as follows: In Section 4.2, we provide a background on RBMs, DBMs, and the ClassRBM approach. Then, in Section 4.3, we present an overview of ClassDBM structure and probability distributions. Afterwards, the training and prediction procedures of the ClassDBM are explained in detail. Section 4.4 provides performance analysis of our method in comparison with existing methods on two widely known benchmark data. Thereafter, the performance of the method on real-world advertising data is evaluated, and insights on model and relationships among modalities is provided. Finally, in Section 5, we conclude and provide future research directions.

4.2 Background

In this chapter, we briefly explain the structures and probability distributions of RBMs and DBMs. Moreover, we explain how each model is trained for generative learning. Afterwards, we review the ClassRBM architecture, probabilities, and training and prediction algorithms. Thereafter, the limitations of ClassRBMs for deeper architecture are illustrated.

4.2.1 Restricted Boltzmann Machines

Restricted Boltzmann Machine (RBM) is a probabilistic model that defines the probability distributions over pairs of visible observations $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and hidden units $\mathbf{h} = (h_1, h_2, \ldots, h_l)$. The joint probability is

$$P(\mathbf{x}, \mathbf{h}; \theta) = \frac{e^{-E(\mathbf{x},\mathbf{h})}}{Z(\theta)}, \tag{4.1}$$

where Z is the normalizing factor, equal to summation over all possible combinations of visible vectors $\mathbf{x}$ and hidden vectors $\mathbf{h}$, i.e. $Z(\theta) = \sum_{\mathbf{x},\mathbf{h}} e^{-E(\mathbf{x},\mathbf{h})}$. And θ is the model parameters, i.e. $= \mathbf{W}, \mathbf{b}, \mathbf{c}$. Also, E is the energy function, described as

$$E(\mathbf{x}, \mathbf{h}) = -\mathbf{h}^T \mathbf{W} x - \mathbf{b}^T \mathbf{x} - \mathbf{c}^T \mathbf{h}, \tag{4.2}$$

where $\mathbf{b}$ is the vector of biases for the visible units, $\mathbf{c}$ is the vector of biases for the hidden units, and $\mathbf{W}$ is the matrix of connection weights. The model is presented in Figure 4.1.

Assuming binary input variables, the conditional probabilities are described in Equation 3.10:

$$
\begin{aligned}
p(\mathbf{x}|\mathbf{h}) &= \prod_i p(x_i|\mathbf{h}) \\
p(x_i = 1|\mathbf{h}) &= \sigma(b_i + \sum_i W_{i,j} h_j) \\
p(\mathbf{h}|\mathbf{x}) &= \prod_i p(h_j|\mathbf{x}) \\
p(h_j = 1|\mathbf{x}) &= \sigma(c_j + \sum_i W_{j,i} x_i),
\end{aligned} \tag{4.3}
$$

where σ is the logistic sigmoid function, i.e. $\sigma(x) = \dfrac{1}{1+e^{-x}}$

The above equations can be moderately adjusted to permit the input vector to take real values. For instance, Gaussian RBMs were developed to model vectors of data with Gaussian distributions, such as images, videos, speech, etc [37, 54]. Moreover, the Replicated SoftMax model is another variant of RBMs to model sparse word count vectors [55]. For modeling simplicity, we focus on binary inputs in this paper, while the proposed methods can be easily extended to real value observations.

Training the model

The probability of observing a visible vector $\mathbf{x}$ is calculated by summing over all values that hidden units can take.

$$
p(\mathbf{x}) = \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{x},\mathbf{h})}}{Z(\theta)} = \frac{\tilde{p}(\mathbf{x})}{Z(\theta)}. \tag{4.4}
$$

However, partition function Z is intractable for all RBMs except for a few networks where sum over all visible and hidden states is feasible to compute [79].

To achieve a generative model for input observations, we wish to maximize the log-likelihood of the probability of observing training data, shown as follows:

$$
\mathcal{L}(X_{train}) = \sum_{i=1}^{|X_{train}|} \log p(\mathbf{x}_i; \theta), \tag{4.5}
$$

where X_{train} is the set of all training observations. Since $p(\mathbf{x}_i)$ is intractable, calculating the gradients is not straightforward. The gradients of $log\ p(\mathbf{x}_i)$ corresponding to the parameters can be decomposed as,

$$\frac{\partial}{\partial\theta}\log p(\mathbf{x}_i;\theta) = \frac{\partial}{\partial\theta}\log \tilde{p}(\mathbf{x}_i;\theta) - \frac{\partial}{\partial\theta}\log Z(\theta), \tag{4.6}$$

where the first expression is called the positive phase and the second expression is called the negative phase of learning. For RBMs, calculating the positive phase is straightforward, but the negative phase is intractable and must be approximated. It is easy to show that the negative and positive phases can be calculated as [44]

$$\frac{\partial}{\partial\theta}\log \tilde{p}(\mathbf{x}_i;\theta) = \mathbf{E}_{\mathbf{h}|\mathbf{x}_i}[\frac{\partial}{\partial\theta}\log \tilde{p}(\mathbf{x}_i,\mathbf{h};\theta)] \tag{4.7}$$

$$\frac{\partial}{\partial\theta}\log Z(\theta) = \mathbf{E}_{\mathbf{x},\mathbf{h}}[\frac{\partial}{\partial\theta}\log \tilde{p}(\mathbf{x},\mathbf{h};\theta)]. \tag{4.8}$$

Hence, using Equation 4.8 we can write Equation 4.6 as

$$\frac{\partial}{\partial\theta}\log p(\mathbf{x}_i;\theta) = \mathbf{E}_{\mathbf{h}|\mathbf{x}_i}[\frac{\partial}{\partial\theta}E(\mathbf{x}_i,\mathbf{h})] - \mathbf{E}_{\mathbf{h},\mathbf{x}}[\frac{\partial}{\partial\theta}E(\mathbf{x},\mathbf{h})]. \tag{4.9}$$

Computing the exact values of the second expression is computationally intractable, hence several works have been introduced to approximate values for the expectation. This includes Contrastive Divergence (CD) [53, 51], Stochastic Maximum Likelihood (SML) [124, 111], recognized as Persistent Contrastive Divergense (PCD) as well, ratio matching [56], etc.

In CD, the expectations of hidden and visible states are approximated using a sample produced after a few iterations of Markov Chain Monte Carlo (MCMC) simulation (Gibbs sampling), where the sample's initial state is set as the visible variable $\mathbf{x}_i$. In [20], the authors showed that using Gibbs sampling with only one iteration, the CD algorithm can

produce a very small bias and fast learning. In PCD, a persistent Gibbs sampling simulation is used to successively sample from the conditions $p(\mathbf{h}|\mathbf{x})$ and $p(\mathbf{x}|\mathbf{h})$. These samples are then used for approximating the negative phase in Equation 4.9.

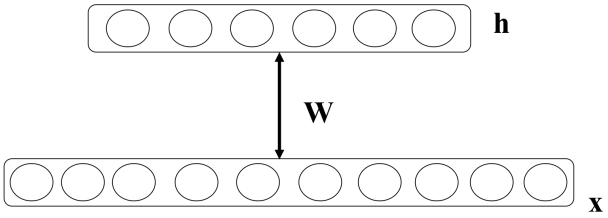


Figure 4.1: RBM structure

4.2.2 Deep Boltzmann Machines

While RBMs assume only one hidden layer, for more complicated structures, Deep Boltzmann Machines show better ability to model the data. The structure of a three-layer Deep Boltzmann Machine is shown in Figure 4.2.

For a DBM with m hidden layers, the energy function is

$$E(\mathbf{x}, \mathbf{h}^1, \mathbf{h}^2, \ldots, \mathbf{h}^m; \theta) = -\mathbf{x}^T \mathbf{W}^1 \mathbf{h}^1 - \mathbf{h}^{1T} \mathbf{W}^2 \mathbf{h}^2 - \cdots - \mathbf{h}^{m-1T} \mathbf{W}^m \mathbf{h}^m, \tag{4.10}$$

where $\theta = \mathbf{W}^1, \mathbf{W}^2, \ldots, \mathbf{W}^m$. Also, the conditional distributions are as follows:

$$\begin{aligned}
p(h_j^1 = 1|\mathbf{x}, \mathbf{h}^2) &= \sigma(\sum_i \mathbf{W}_{ij}^1 x_i + \sum_k \mathbf{W}_{jk}^2 h_k^2) \\
p(h_p^l = 1|\mathbf{h}^{l-1}, \mathbf{h}^{l+1}) &= \sigma(\sum_i \mathbf{W}_{ip}^l h_i^{l-1} + \sum_k \mathbf{W}_{pk}^{l+1} h_k^{l+1}) \quad 1 < l < m \\
p(h_k^m = 1|\mathbf{h}^{m-1}) &= \sigma(\sum_i \mathbf{W}_{im}^2 h_i^{m-1}) \\
p(x_i = 1|\mathbf{h}^1) &= \sigma(\sum_j \mathbf{W}_{ij}^1 h_j^1).
\end{aligned} \tag{4.11}$$

Training the model

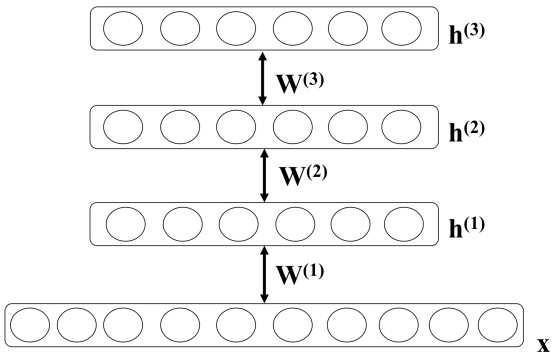


Figure 4.2: A 3 layer DBM structure

In DBMs, the probability of observing an observation vector **x** is calculated as

$$p(\mathbf{x};\theta) = \frac{1}{Z(\theta)} \sum_{\{\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m\}} e^{-E(\mathbf{x},\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m;\theta)} = \frac{\tilde{p}(\mathbf{x},\theta)}{Z(\theta)}. \tag{4.12}$$

The DBM energy function in Equation 4.10 includes connection between hidden units. These connections significantly affect the model by causing the positive phase in Equation 4.17 to be intractable. This is because, by having more than one layer, all the combinations of states in the different layers must be calculated in the positive phase. However, as the number of observations in each layer grows, computing combinations of all states is impractical. In the DBMs model, we wish to maximize the log-likelihood of train data probability, similar to RBMs in Equation 4.9:

$$\begin{aligned}\frac{\partial}{\partial\theta}\log p(\mathbf{x}_i;\theta) &= \frac{\partial}{\partial\theta}\log\tilde{p}(\mathbf{x}_i;\theta) - \frac{\partial}{\partial\theta}\log Z(\theta) \\ &= \mathbf{E}_{\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m|\mathbf{x}_i}[\frac{\partial}{\partial\theta}E(\mathbf{x}_i,\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m)] - \mathbf{E}_{\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m,\mathbf{x}}[\frac{\partial}{\partial\theta}E(\mathbf{x},\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m)].\end{aligned} \tag{4.13}$$

Here, to calculate the positive term (the first expression in Equation 4.13), we need to compute $\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m|\mathbf{x}$; however, because of the interactions between layers, the joint probability of hidden layers given input is not separable. Therefore, the positive phase is intractable and needs to be approximated for training. An approximation method com-

monly used for the positive part is the mean field inference method [100]. The mean field approximation is a simple form of variational inference. In the variational approximation methods, we wish to approximate a specific distribution by some appropriate simple family distributions where the approximating distribution is assumed to be fully factorial. This method can properly capture bidirectional interactions between layers. In the mean field approximation, we assume that the approximating distribution is fully factorial.

4.2.3 Classification RBM (ClassRBM)

The Classification RBM (ClassRBM) was introduced for expanding RBMs into discriminative problems [68]. The Structure of ClassRBM is similar to that of regular RBMs, however; the visible layer includes both the observations and their corresponding class labels. The structure is shown in Figure 4.3.

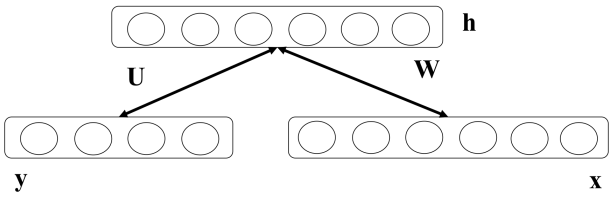


Figure 4.3: Class RBM

Assuming the visible observation $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, the class label as $\mathbf{y}$, and the hidden units as $\mathbf{h} = (h_1, h_2, \ldots, h_m)$, the joint probability distribution and energy function takes the form of

$$p(\mathbf{x}, \mathbf{y}, \mathbf{h}; \theta) = \frac{e^{-E(\mathbf{x},\mathbf{y},\mathbf{h})}}{Z(\theta)}$$
$$E(\mathbf{x}, \mathbf{y}, \mathbf{h}) = -\mathbf{h}^T \mathbf{W} x - \mathbf{b}^T \mathbf{x} - \mathbf{c}^T \mathbf{h} - \mathbf{d}^T \mathbf{y} - \mathbf{h}^T \mathbf{U} \mathbf{y}, \tag{4.14}$$

where Z is equal to summation over all values that hidden units can take. $\mathbf{b}$ is the vector of biases for the visible units, $\mathbf{c}$ is the vector of biases for the hidden units, $\mathbf{d}$ is the vector of

biases for class labels, **W** is the matrix of connection weights between visible observations and hidden units, and **U** is the matrix of connection weights between class labels and hidden units. Moreover, θ refers to the model parameters equal to $(\mathbf{W}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{U})$, and E is the energy function.

In this model, the conditional distributions will be as follows:

$$\begin{aligned} p(\mathbf{x}|\mathbf{h}) &= \prod_i p(x_i|\mathbf{h}) \\ p(x_i = 1|\mathbf{h}) &= \sigma(b_i + \sum_j W_{ji} h_j) \\ p(y_l = 1|\mathbf{h}) &= \frac{e^{d_l + \sum_j U_{jl} h_j}}{\sum_{y^*} e^{d_{y*} + \sum_j U_{jy*} h_j}}. \\ p(\mathbf{h}|\mathbf{y}, \mathbf{x}) &= \prod_j p(h_j|\mathbf{y}, \mathbf{x}) \\ p(h_j = 1|\mathbf{y}, \mathbf{x}) &= \sigma(c_j + U_{jy} + \sum_i W_{ji} x_i) \end{aligned} \tag{4.15}$$

Equation 4.15 shows that the hidden units are capable of representing information of the visible observations as well as the class labels simultaneously.

For training a generative model, assuming that the training data is $D_{train} = (\mathbf{x}_i, \mathbf{y}_i)$ for $i = 1, ..., n$,

$$p(\mathbf{x}, \mathbf{y}) = \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{x}, \mathbf{y}, \mathbf{h})}}{Z} = \frac{\tilde{p}(\mathbf{x}, \mathbf{y})}{Z}. \tag{4.16}$$

Hence, the log-likelihood for a generative model will be

$$\mathcal{L}_{gen}(D_{train}) = \sum_{i=1}^{|D_train|} log \;\; p(y_i, x_i). \tag{4.17}$$

To maximize the log-likelihood (eq. 4.17), its gradients can be estimated using

$$\frac{\partial \log p(y_i, x_i)}{\partial \theta} = \mathbf{E}_{\mathbf{h}|y_i, x_i}[\frac{\partial}{\partial \theta} E(y_i, x_i, \mathbf{h})] - \mathbf{E}_{\mathbf{y}, \mathbf{x}, \mathbf{h}}[\frac{\partial}{\partial \theta} E(\mathbf{y}, \mathbf{x}, \mathbf{h})]. \tag{4.18}$$

In this generative model, similar to regular RBMs, the first expression is tractable while the second one is intractable. Hence, the intractable expression can be approximated by Gibbs sampling (Contrastive Divergence).

For a discriminative model, instead of the joint distribution of observations and class labels, $p(\mathbf{y}, \mathbf{x})$, the authors suggest to minimize the conditional distribution of class labels given observations, i.e. $p(\mathbf{y}|\mathbf{x})$. The authors showed that $p(\mathbf{y}|\mathbf{x})$ can be calculated directly. This is based on the fact that the values that $\mathbf{y}$ can take are limited in classification problems. Therefore, conditional distribution can be directly calculated as

$$p(\mathbf{y}|\mathbf{x}) = \frac{e^{d_y} \prod_{j=1}^{n} (1 + e^{c_j + U_{jy} + \sum_i W_{ji} x_i})}{\sum_{y^*} e^{d_{y^*}} \prod_{j=1}^{n} (1 + e^{c_j + U_{jy^*} + \sum_i W_{ji} x_i})}. \tag{4.19}$$

Thus, in discriminative RBM, the likelihood function can be shown as

$$\mathcal{L}_{disc}(D_{train}) = \sum_{i}^{|D_{train}|} \log p(\mathbf{y}_i|\mathbf{x}_i). \tag{4.20}$$

[68] showed that ClassRBM can be trained by stochastic gradient approximation, where the exact gradients are computed as follows:

$$\begin{aligned} \frac{\partial log p(y_i|x_i)}{\partial \theta} &= \sum_j \sigma(o_{yj}(x_i)) \frac{\partial o_{yj}(x_i)}{\partial \theta} \\ &\quad - \sum_{j,y*} \sigma(o_{y*j}(x_i)) p(y^*|x_i) \frac{\partial o_{y*j}(x_i)}{\partial \theta}, \end{aligned} \tag{4.21}$$

where $o_{yj}(x) = c_j + \sum_k W_{jk} x_k + U_{jy}$.

4.3 Methodology Overview

In the previous section, we reviewed the ClassRBM methodology. In ClassRBM, the conditional distributions can be directly calculated when the number of classes is limited. This

characteristic allows us to learn the exact gradients using stochastic gradient descent methods. In this section, we introduce ClassDBM approach, which is an expansion of ClassRBMs to deeper architectures. We show that when extending to deeper architecture, the gradients are not straight-forward to calculate. Hence, we present a new training and prediction algorithm for ClassDBMs. Moreover, we illustrate that this method can be expanded to models with more than one modality.

4.3.1 Classification DBM (ClassDBM)

In this section, we wish to extend the ClassRBM to a deeper architecture where there exist more than one hidden layer. We name this new architecture as Classification DBM (ClassDBM). We can treat a ClassDBM as an expansion of ClassRBM to architectures with more than one hidden layer. One difficulty when learning ClassDBMs is that the excess layers will cause the calculations of gradients to be intractable, and the gradients need to be approximated. Hence, solving a ClassDBM is not as straightforward as a SlassRBM. To represent a ClassDBM structure, let's assume the number of hidden layers is two. In this case, the energy function and joint probability will be.

$$E(\mathbf{y}, \mathbf{x}, \mathbf{h}^1, \mathbf{h}^2; \theta) = -\mathbf{h}^{1T}\mathbf{W}^1\mathbf{x} - \mathbf{b}^T\mathbf{x} - \mathbf{c}^{1T}\mathbf{h}^1 - \mathbf{d}^T\mathbf{y} - \mathbf{h}^{1T}\mathbf{U}\mathbf{y} - \mathbf{h}^{2T}\mathbf{W}^2\mathbf{h}^1 - \mathbf{c}^{2T}\mathbf{h}^2 \tag{4.22}$$

$$p(\mathbf{y}, \mathbf{x}, \mathbf{h}^1, \mathbf{h}^2; \theta) = \frac{exp(-E(\mathbf{y}, \mathbf{x}, \mathbf{h}^1, \mathbf{h}^2))}{Z(\theta)} = \frac{\tilde{p}(\mathbf{y}, \mathbf{x}, \mathbf{h}^1, \mathbf{h}^2; \theta)}{Z(\theta)} \tag{4.23}$$

Where Z is a normalization constant which guarantees that equation 4.23 is a valid probability distribution, and $\mathbf{y}$ is the class label vector such that if the class label is c, then all elements are zero except for the c^{th} element which takes the value of one.

It is easy to show that the condition probabilities will be

$$
\begin{aligned}
&p(\mathbf{x}|\mathbf{h}^{(1)}) = \prod_i p(x_i|\mathbf{h}^{(1)}) \\
&p(x_i = 1|\mathbf{h}^1) = \sigma(b_i + \sum_j W^1_{j,i} h^1_j) \\
&p(\mathbf{h}^{(1)}|\mathbf{y}, \mathbf{x}, \mathbf{h}^2) = \prod_j p(h^1_j|\mathbf{y}, \mathbf{x}, \mathbf{h}^2) \\
&p(h^1_j = 1|\mathbf{y}, \mathbf{x}, \mathbf{h}^2) = \sigma(c_j + \sum_p U_{jy} + \sum_i W^1_{j,i} x^i + \sum_l W^2_{l,i} h^2_{k,l}) \\
&p(\mathbf{h}^2|\mathbf{h}^1) = \prod_j p(h^2_j|\mathbf{h}^1_1) \\
&p(h^2_j = 1|\mathbf{h}^1_1) = \sigma(d_j + \sum_l W^2_{1_{l,j}} h^1_l) \\
&p(y_l = 1|\mathbf{h}^1) = \frac{e^{d_l + \sum_j U_{jl} h^1_j}}{\sum_{y^*} e^{d_{y*} + \sum_j U_{jy*} h^1_j}}
\end{aligned}
\tag{4.24}
$$

Here, first we will show that in spite of ClassRBMs, the extension to deeper architectures is not tractable. To do this, by inspiring from [68] and by taking into account that class labels are limited, we will show that $p(\mathbf{y}|\mathbf{x})$ can be written as,

$$
p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{y}, \mathbf{x})}{p(\mathbf{x})} = \frac{\tilde{p}(\mathbf{y}, \mathbf{x})}{\tilde{p}(\mathbf{x})} \tag{4.25}
$$

Where,

$$
\begin{aligned}
\tilde{p}(\mathbf{y}, \mathbf{x}) &= \sum_{\mathbf{h}^2} \sum_{\mathbf{h}^1} \tilde{p}(\mathbf{x}, \mathbf{y}, \mathbf{h}^1, \mathbf{h}^2) \\
&= \sum_{\mathbf{h}^2} \sum_{\mathbf{h}^1} exp(\mathbf{h}^{1T} \mathbf{W}^1 \mathbf{x} + \mathbf{b}^T \mathbf{x} + \mathbf{c}^{1T} \mathbf{h}^1 + \mathbf{d}^T \mathbf{y} + \mathbf{h}^{1T} \mathbf{U} \mathbf{y} + \mathbf{h}^{2T} \mathbf{W}^2 \mathbf{h}^1 + \mathbf{c}^{2T} \mathbf{h}^2)
\end{aligned}
\tag{4.26}
$$

Assuming that first hidden layer has n_1 units, second hidden layer has n_2 units, $p(\mathbf{y}, \mathbf{x})$ can be expended as,

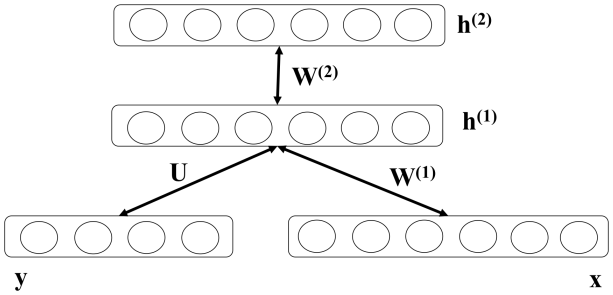


Figure 4.4: Class DBM

$$
\begin{aligned}
p(y=l,\mathbf{x}) &= \sum_{h_1^2,\ldots,h_{n_2}^2\in\{0,1\}} \left(\sum_{h_1^1,\ldots,h_{n_1}^1\in\{0,1\}} exp(\sum_j \sum_i w_{j,i}^1 x_i h_j^1 + \sum_j c_j^1 h_j^1 + d_l + \sum_j U_{jl} h_j^1 \right. \\
&\qquad \left. \sum_j \sum_k w_{k,j}^2 h_k^2 h_j^1 + \sum_k c_k^2 h_k^2) \right) \\
&= exp(d_l) \sum_{h_1^2,\ldots,h_{n_2}^2\in\{0,1\}} \left(exp(\sum_k c_k^2 h_k^2) \prod_j (1 + exp(\sum_i w_{j,i}^1 x_i + c_j^1 + U_{jl} + \sum_k w_{kj}^2 h_k^2)) \right) \\
&= exp(d_l) \sum_{h_1^2,\ldots,h_{n_2}^2\in\{0,1\}} \left(exp(\sum_k c_k^2 h_k^2 + \sum_j log(1 + exp(\sum_i w_{j,i}^1 x_i + c_j^1 + U_{jl} + \sum_k w_{kj}^2 h_k^2))) \right)
\end{aligned}
\tag{4.27}
$$

Assuming higher hidden layers have no bias terms, we will have

$$
\begin{aligned}
p(y=l,\mathbf{x}) &= \sum_{h_1^2,\ldots,h_{n_2}^2\in\{0,1\}} \left(exp(d_l + \sum_j \log\left(1 + exp(\sum_i w_{j,i}^1 x_i + c_j^1 + U_{jl} + \sum_k w_{kj}^2 h_k^2)\right)) \right) \\
&= \sum_{h_1^2,\ldots,h_{n_2}^2\in\{0,1\}} e^{d_l} \prod_{j=1}^{n_1} (1 + e^{c_j^1 + U_{jl} + \sum_i W_{ji}^1 x_i + \sum_k w_{kj}^2 h_k^2})
\end{aligned}
\tag{4.28}
$$

Given that number of class labels, n_c, are limited, we can compute the $p(\mathbf{x})$ as,

$$\begin{aligned} p(\mathbf{x}) &= \sum_{l^*=1}^{n_c} p(y=l^*, \mathbf{x}) \\ &= \sum_{h_1^2,\dots,h_{n_2}^2 \in \{0,1\}} \sum_{l^*=1}^{n_c} e^{d_{l^*}} \prod_{j=1}^{n_1} (1 + e^{c_j^1 + U_{jl^*} + \sum_i W_{ji}^1 x_i + \sum_k w_{kj}^2 h_k^2}) \end{aligned} \tag{4.29}$$

Hence, the conditional probability can be written as

$$\tilde{p}(y=l|\mathbf{x}) = \frac{\sum_{h_1^2,\dots,h_{n_2}^2 \in \{0,1\}} e^{d_l} \prod_{j=1}^{n_1} (1 + e^{c_j^1 + U_{jl} + \sum_i W_{ji}^1 x_i + \sum_k w_{kj}^2 h_k^2})}{\sum_{h_1^2,\dots,h_{n_2}^2 \in \{0,1\}} \sum_{l^*=1}^{n_c} e^{d_{l^*}} \prod_{j=1}^{n_1} (1 + e^{c_j^1 + U_{jl^*} + \sum_i W_{ji}^1 x_i + \sum_k w_{kj}^2 h_k^2})}. \tag{4.30}$$

Here, in spite of conditional distribution in equation 4.19 for ClassRBMs, this conditional distribution is intractable. This is because both nominator and denominator has $\sum_{\mathbf{h}^{(2)}}$. Note that this sum will be even more hard to calculate if the number of hidden layers is more than two. Therefore, the learning algorithm presented for ClassRBM cannot be applied to the ClassDBM approach. Hence to find the gradients, we need to use approximate methods. In the following subsection, we will present a learning algorithm to approximate the gradients.

4.3.2 Training Model

For training the model, we are aiming at minimizing the negative log likelihood of $p(\mathbf{y}|\mathbf{x})$.

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{x})} = \frac{\tilde{p}(\mathbf{x},\mathbf{y})/Z(\theta)}{\tilde{p}(\mathbf{x})/Z(\theta)} = \frac{\tilde{p}(\mathbf{x},\mathbf{y})}{\sum_{y^*=1}^{n_c} \tilde{p}(\mathbf{x},\mathbf{y}^*)}$$

Hence, we can write the log-likelihood as,

$$\mathcal{L}(D_{train}) = \sum_{i=1}^{D_{train}} \log p(\mathbf{y}_i|\mathbf{x}_i)$$
$$\log p(\mathbf{y}|\mathbf{x}) = \log \tilde{p}(\mathbf{x},\mathbf{y}) - log \sum_{y^*=1}^{n_c} \tilde{p}(\mathbf{x},\mathbf{y}^*) \tag{4.31}$$

The first expression is similar to positive expression in DBMs in equation 4.13. Derivative of the first expression can be written as,

$$\begin{aligned}
\frac{\partial}{\partial\theta}\log\tilde{p}(\mathbf{x},\mathbf{y}) &= \frac{\partial}{\partial\theta}\log\sum_{\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m}\tilde{p}(\mathbf{x},\mathbf{y},\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m)\\
&= \frac{1}{\tilde{p}(\mathbf{x},\mathbf{y})}[\frac{\partial}{\partial\theta}\sum_{\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m}\tilde{p}(\mathbf{x},\mathbf{y},\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m)]\\
&= \frac{1}{\tilde{p}(\mathbf{x},\mathbf{y})}[\sum_{\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m}\frac{\partial}{\partial\theta}\tilde{p}(\mathbf{x},\mathbf{y},\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m)]\\
&= \frac{1}{\tilde{p}(\mathbf{x},\mathbf{y})}[\sum_{\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m}\frac{\partial}{\partial\theta}e^{log(\tilde{p}(\mathbf{x},\mathbf{y},\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m))}]\\
&= \frac{1}{\tilde{p}(\mathbf{x},\mathbf{y})}[\sum_{\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m}\tilde{p}(\mathbf{x},\mathbf{y},\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m)\frac{\partial}{\partial\theta}\log\tilde{p}(\mathbf{x},\mathbf{y},\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m)]\\
&= \sum_{\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m}\frac{\tilde{p}(\mathbf{x},\mathbf{y},\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m)}{\tilde{p}(\mathbf{x},\mathbf{y})}\frac{\partial}{\partial\theta}\log\tilde{p}(\mathbf{x},\mathbf{y},\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m)\\
&= \sum_{\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m}p(\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m|\mathbf{x},\mathbf{y})\frac{\partial}{\partial\theta}\log\tilde{p}(\mathbf{x},\mathbf{y},\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m)
\end{aligned} \tag{4.32}$$

Hence,

$$\frac{\partial}{\partial\theta}log\ \tilde{p}(\mathbf{x},\mathbf{y}) = \mathbf{E}_{\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^{(m)}|\mathbf{x},\mathbf{y}}[\frac{\partial}{\partial\theta}E(\mathbf{x},\mathbf{y},\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m)] \tag{4.33}$$

This derivative can be calculated using mean field inference. Moreover, we can expand the second expression in equation 4.31 as,

$$
\begin{aligned}
\frac{\partial}{\partial\theta}\log\sum_{y^*=1}^{n_c}\tilde{p}(\mathbf{x},\mathbf{y}^*) &= \frac{1}{\sum_{y^*=1}^{n_c}\tilde{p}(\mathbf{x},\mathbf{y}^*)}\frac{\partial}{\partial\theta}\sum_{y^*=1}^{n_c}\tilde{p}(\mathbf{x},\mathbf{y}^*) \\
&= \frac{1}{\sum_{y^*=1}^{n_c}\tilde{p}(\mathbf{x},\mathbf{y}^*)}\sum_{y^*=1}^{n_c}\frac{\partial}{\partial\theta}\tilde{p}(\mathbf{x},\mathbf{y}^*) \\
&= \frac{1}{\sum_{y^*=1}^{n_c}\tilde{p}(\mathbf{x},\mathbf{y}^*)}\sum_{y^*=1}^{n_c}\frac{\partial}{\partial\theta}exp(\log\tilde{p}(\mathbf{x},\mathbf{y}^*)) \\
&= \frac{1}{\sum_{y^*=1}^{n_c}\tilde{p}(\mathbf{x},\mathbf{y}^*)}\sum_{y^*=1}^{n_c}exp(\log\tilde{p}(\mathbf{x},\mathbf{y}^*))\frac{\partial}{\partial\theta}\log\tilde{p}(\mathbf{x},\mathbf{y}^*) \\
&= \sum_{y^*=1}^{n_c}\frac{\tilde{p}(\mathbf{x},\mathbf{y}^*)}{\tilde{p}(\mathbf{x})}\frac{\partial}{\partial\theta}\log\tilde{p}(\mathbf{x},\mathbf{y}^*) \\
&= \sum_{y^*=1}^{n_c}\frac{\sum_{\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m}\tilde{p}(\mathbf{x},\mathbf{y}^*,\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m)}{\tilde{p}(\mathbf{x})}\frac{\partial}{\partial\theta}\log\tilde{p}(\mathbf{x},\mathbf{y}^*) \\
&= \mathbf{E}_{\mathbf{y},\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m|\mathbf{x}}\frac{\partial}{\partial\theta}E(\mathbf{x},\mathbf{y},\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m)
\end{aligned}
\tag{4.34}
$$

Hence,

$$
\begin{aligned}
\frac{\partial}{\partial\theta}\log p(\mathbf{y}_i|\mathbf{x}_i) = &\mathbf{E}_{\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m|\mathbf{x}_i,\mathbf{y}_i}[\frac{\partial}{\partial\theta}E(\mathbf{x}_i,\mathbf{y}_i,\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m)] \\
&- \mathbf{E}_{\mathbf{y},\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m|\mathbf{x}_i}[\frac{\partial}{\partial\theta}E(\mathbf{x}_i,\mathbf{y},\mathbf{h}^1,\mathbf{h}^2,\ldots,\mathbf{h}^m)]
\end{aligned}
\tag{4.35}
$$

To find the gradients in equation 4.35, we need to approximate the gradients expectations for each expression. For the first expression, we suggest to use mean field (MF) approximation to estimate the gradient. To perform MF, for each observation i, we will find the mean expectation of each layer given $\mathbf{x}_i$, and $\mathbf{y}_i$. Also, to approximate the expectation in second equation, we suggest using Gibbs Sampling method to sample from $\mathbf{y}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \ldots, \mathbf{h}^{(m)}$ observations given $\mathbf{x}$.

Algorithm 4.1 Training ClassDBM

Initializing the parameters
Set α the learning rate, β momentum, s maximum number of mean field steps, k the number of Gibbs steps, m the number of minibatch data
Initialize the matrices $\tilde{\mathbf{H}}^1_{p\times n_1}$, $\tilde{\mathbf{H}}^2_{p\times n_2}$, to random values from Bernoulli distribution
while not converged (learning loop) **do**
    sample a batch of n examples from training data
    arrange samples as $\mathbf{X}_{m\times p}$ and $\mathbf{Y}_{m\times n_c}$
    initialize matrices $\hat{\mathbf{H}}^1_{p\times n_1}$, $\hat{\mathbf{H}}^2_{p\times n_2}$ to the model marginals
    while Mean Field inference loop not converged **do**
        $\hat{\mathbf{H}}^1 \leftarrow \sigma(\mathbf{X}\mathbf{W}^1 + \mathbf{Y}\mathbf{U} + \hat{\mathbf{H}}^2\mathbf{W}^{2^T})$
        $\hat{\mathbf{H}}^2 \leftarrow \sigma(\hat{\mathbf{H}}^1\mathbf{W}^2)$
    end while
    $\delta^+\mathbf{W}^1 \leftarrow \frac{1}{n}\mathbf{X}^T\hat{\mathbf{H}}^1$
    $\delta^+\mathbf{U} \leftarrow \frac{1}{n}\mathbf{Y}^T\hat{\mathbf{H}}^1$
    $\delta^+\mathbf{W}^2 \leftarrow \frac{1}{n}\hat{\mathbf{H}}^1\hat{\mathbf{H}}^{2T}$
    while Gibbs sample number is not reached **do**
        $\tilde{\mathbf{Y}} \leftarrow \mathbf{Y}$
        for $j = 1$ `to` k **do**
            $\tilde{\mathbf{H}}^1 \leftarrow$ *sample ftom* $\sigma((\mathbf{X}\mathbf{W}^1 + \tilde{\mathbf{Y}}\mathbf{U} + \tilde{\mathbf{H}}^2\mathbf{W}^{2^T})$
            $\tilde{\mathbf{H}}^2 \leftarrow$ *sample ftom* $\sigma(\tilde{\mathbf{H}}^1\mathbf{W}^2)$
            $\tilde{\mathbf{Y}} \leftarrow$ *sample ftom* $softmax(\tilde{\mathbf{H}}^1\mathbf{U}^T)$
        end for
    end while
    $\delta^-\mathbf{W}^1 \leftarrow \frac{1}{n}\mathbf{X}^T\tilde{\mathbf{H}}^1$
    $\delta^-\mathbf{U} \leftarrow \frac{1}{n}\tilde{\mathbf{Y}}^T\tilde{\mathbf{H}}^1$
    $\delta^-\mathbf{W}^2 \leftarrow \frac{1}{n}\tilde{\mathbf{H}}^1\tilde{\mathbf{H}}^{2T}$
    Updating Parameters
    $\mathbf{W}^1 \leftarrow \mathbf{W}^1 + \alpha(\delta^+\mathbf{W}^1 - \delta^-\mathbf{W}^1)$
    $\mathbf{W}^2 \leftarrow \mathbf{W}^2 + \alpha(\delta^+\mathbf{W}^2 - \delta^-\mathbf{W}^2)$
    $\mathbf{U} \leftarrow \mathbf{U} + \alpha(\delta^+\mathbf{U} - \delta^-\mathbf{U})$
end while

4.3.3 Inference Model

After learning the model parameters with the method explained in section 4.3.2, we need to have a procedure for predicting class labels of new observations. In a ClassRBM $p(\mathbf{y}|\mathbf{x})$ can be directly calculated using equation 4.19. However, in equation 4.30, we showed that this probability is intractable when the number of hidden layers is larger than one. To calculate the exact probabilities, we need to have the values of all hidden layers in the model. In order to do this, for each new observation, we suggest approximating the expectation of the hidden values for each class label. Afterward, we can use these approximations to calculate the probability of observing each class label. Hence, we can select the label with the highest probability as our prediction. To approximate the expectation values of hidden layers given observations $\mathbf{x}_i$ and each class label y^*, we use mean field approximation. The prediction algorithm is explained in algorithm 4.2

Algorithm 4.2 Predicting a new observation in ClassDBM

1: Set the weights $\mathbf{U}$, $\mathbf{W}^1$, and $\mathbf{W}^2$ equal to the weights learned in algorithm 4.1.
2: **for** observation i in $1, \ldots, m$ **do**
3: Set the new observation as $\mathbf{x}_i$
4: **for** class label c in $1, \ldots, n_c$ **do**
5: $\mathbf{y}_c$ as a $n_c \times 1$ vector, where all values are zeros except for c^{th} entry which is equal to one.
6: Run mean field approximation similar to line 8 to 11 in algorithm 4.1 to approximate values of $\hat{\mathbf{h}}^1$ and $\hat{\mathbf{h}}^2$
7: calculate the negative entropy $\mathbf{e}[c]$ as $E(\mathbf{x}_i, \mathbf{y}_c, \hat{\mathbf{h}}^1, \hat{\mathbf{h}}^2)$
8: **end for**
9: $\hat{\mathbf{y}}_i = argmax(\mathbf{e})$
10: **end for**

4.3.4 Multi Modal Class DBM

In this section, we extend the ClassDBM methodology to cases with more than one modality. In real-world, observations are normally stored in different forms of modality. For example, images can be accomponied with their corresponding captions and tags, videos

are combination of visual and audio information, speech audio files can be accomponied with the speech script, and etc. Therefore, it is beneficiary to have a model that takes into account all modalities. In [107], the authors presented a novel DBM structure to take into account multiple modalities. The structure of a multimodal DBM with two modalities and three hidden layers is presented in figure 4.5 (a). In this model, the authors assume that each modality i (i in $1, 2, \ldots, q$) is connected to their corresponding hidden layer with a specific weight matrix. However, the last hidden layer is a joint representation of previous layers. To learn the model, the authors propose to use mean-field inference to approximate the expectations of hidden units given data, and Gibbs sampling to estimate the model expectations.

To learn the multimodal ClassDBM, inspiring from the ClassDBM approach, we assume that class label vector $\mathbf{y}$ is connected to hidden layers corresponding to modality i with corresponding weight matrix $\mathbf{U}^i$. The structure for two modalities and three hidden layers is shown in figure 4.5 (b).

For this structure, the energy function and joint probability will be

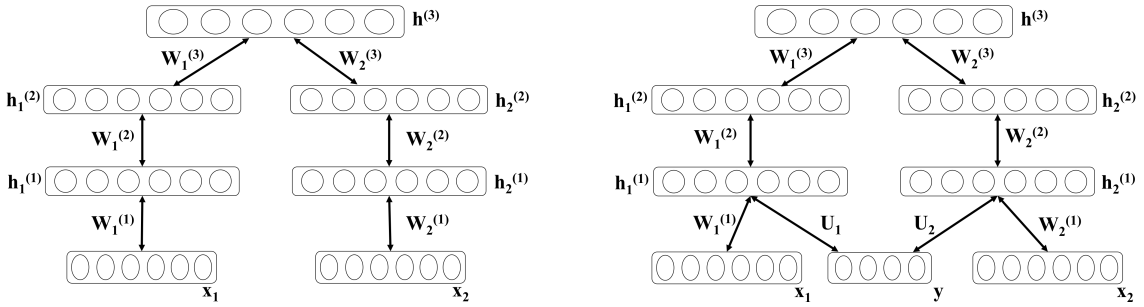
$$\begin{aligned} E(\mathbf{y}, \mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \mathbf{h}^1_{(1)}, \mathbf{h}^1_{(2)}, \mathbf{h}^2_{(1)}, \mathbf{h}^2_{(2)}, \mathbf{h}^3; \theta) = & -{\mathbf{h}^1_{(1)}}^T \mathbf{W}^1_{(1)} \mathbf{x}_{(1)} - {\mathbf{h}^1_{(2)}}^T \mathbf{W}^1_{(2)} \mathbf{x}_{(2)} - \mathbf{b}^T_{(1)} \mathbf{x}_{(1)} \\ & - \mathbf{b}^T_{(2)} \mathbf{x}_{(2)} - {\mathbf{c}^1_{(1)}}^T \mathbf{h}^1_{(1)} - {\mathbf{c}^1_{(2)}}^T \mathbf{h}^1_{(2)} - \mathbf{d}^T \mathbf{y} \\ & - {\mathbf{h}^1_{(1)}}^T \mathbf{U}_{(1)} \mathbf{y} - {\mathbf{h}^1_{(2)}}^T \mathbf{U}_{(2)} \mathbf{y} - {\mathbf{h}^2_{(1)}}^T \mathbf{W}^2_{(1)} \mathbf{h}^1_{(1)} \\ & - {\mathbf{h}^2_{(2)}}^T \mathbf{W}^2_{(2)} \mathbf{h}^1_{(2)} - {\mathbf{c}^2_{(1)}}^T \mathbf{h}^2_{(1)} - {\mathbf{c}^2_{(2)}}^T \mathbf{h}_{(2)} \\ & - \mathbf{h}^{3T} \mathbf{W}^3_{(1)} \mathbf{h}^2_{(1)} - \mathbf{h}^{3T} \mathbf{W}^3_{(2)} \mathbf{h}^2_{(2)} \end{aligned} \tag{4.36}$$

$$\begin{aligned} p(\mathbf{y}, \mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \mathbf{h}^1_{(1)}, \mathbf{h}^1_{(2)}, \mathbf{h}^2_{(1)}, \mathbf{h}^2_{(2)}, \mathbf{h}^3; \theta) &= \frac{exp(-E(\mathbf{y}, \mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \mathbf{h}^1_{(1)}, \mathbf{h}^1_{(2)}, \mathbf{h}^2_{(1)}, \mathbf{h}^2_{(2)}, \mathbf{h}^3; \theta))}{Z(\theta)} \\ &= \frac{\tilde{p}(\mathbf{y}, \mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \mathbf{h}^1_{(1)}, \mathbf{h}^1_{(2)}, \mathbf{h}^2_{(1)}, \mathbf{h}^2_{(2)}, \mathbf{h}^3; \theta)}{Z(\theta)}. \end{aligned} \tag{4.37}$$

Similar to ClassDBM, the probabilities can be calculated as

$$
\begin{aligned}
&p(\mathbf{x}_k|\mathbf{h}_k^{(1)}) = \prod_i p(x_{k,i}|\mathbf{h}_k^{(1)}) \quad k = 1, 2 \\
&p(x_{k,i} = 1|\mathbf{h}_k^{(1)}) = \sigma(b_{k,i} + \sum_j W_{k_{j,i}}^{(1)} h_{k,j}^{(1)}) \\
&p(\mathbf{h}_k^{(1)}|\mathbf{y}, \mathbf{x}_k, \mathbf{h}_k^{(2)}) = \prod_j p(h_{k,j}^{(1)}|\mathbf{y}, \mathbf{x}_k, \mathbf{h}_k^{(2)}) \\
&p(h_{k,j}^{(1)} = 1|\mathbf{y}, \mathbf{x}_k, \mathbf{h}_k^{(2)}) = \sigma(c_{k,j} + \sum_p U_{k_{j,p} y_p} + \sum_i W_{k_{j,i}} x_k^i + \sum_l W_{k,li} h_{k,l}^{(2)}) \\
&p(\mathbf{h}^{(3)}|\mathbf{h}_1^{(2)}, \mathbf{h}_2^{(2)}) = \prod_j p(h_j^{(3)}|\mathbf{h}_1^{(2)}, \mathbf{h}_2^{(2)}) \\
&p(h_j^{(3)} = 1|\mathbf{h}_1^{(2)}, \mathbf{h}_2^{(2)}) = \sigma(d_j + \sum_l W_{1_{l,j}}^{(2)} h_{1,l}^{(2)} + \sum_r W_{2_{r,j}}^{(2)} h_{2,r}^{(2)}).
\end{aligned}
\tag{4.38}
$$

$$
p(\mathbf{y}|\mathbf{h}_1^{(1)}, \mathbf{h}_2^{(2)}) = \frac{e^{d_y + \sum_j U_{1_{j,y}} h_{1,j}^{(1)} + \sum_j U_{2_{j,y}} h_{1,j}^{(2)}}}{\sum_{y^*=1}^{n_c} e^{d_{y^*} + \sum_j U_{j,y^*}^1 h_{1,j}^1 + \sum_j U_{j,y^*}^2 h_{2,j}^1}}. \tag{4.39}
$$



(a) Multi-modal DBM [107]

(b) Multi-modal ClassDBM

Figure 4.5: Network Structure for Multimodal DBM and ClassDBM (two modalities)

Alike ClassDBM, we wish to maximize the conditional probability $p(\mathbf{y}|\mathbf{x}_1, \mathbf{x}_2)$. The

conditional probability and log likelihood can be written as

$$\mathcal{L} = \sum_{i=1}^{\mathbf{D}_{train}} \log p(\mathbf{y}_i|\mathbf{x}_{i(1)}, \mathbf{x}_{i(2)})$$

$$p(\mathbf{y}|\mathbf{x}_{(1)}, \mathbf{x}_{(2)}) = \frac{p(\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \mathbf{y})}{p(\mathbf{x}_{(1)}, \mathbf{x}_{(2)})} = \frac{\tilde{p}(\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \mathbf{y})}{\sum_{y^*=1}^{n_c} \tilde{p}(\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \mathbf{y}^*)}$$

$$\log p(\mathbf{y}|\mathbf{x}_{(1)}, \mathbf{x}_{(2)}) = \log \tilde{p}(\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \mathbf{y}) - \log \sum_{y^*=1}^{n_c} \tilde{p}(\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \mathbf{y}^*).$$

Hence, we can show that derivatives of log likelihood will be equal to

$$\begin{aligned}\frac{\partial}{\partial\theta} \log p(\mathbf{y}_i|\mathbf{x}_{i(1)}, \mathbf{x}_{i(2)}) =& \mathbf{E}_{\mathbf{h}^1_{(1)},\mathbf{h}^1_{(2)},\mathbf{h}^2_{(1)},\mathbf{h}^2_{(2)},\mathbf{h}^3|\mathbf{x}_{i(1)},\mathbf{x}_{i(2)},\mathbf{y}_i}[\frac{\partial}{\partial\theta} E(\mathbf{y}_i, \mathbf{x}_{i(1)}, \mathbf{x}_{i(2)}, \mathbf{h}^1_{(1)}, \mathbf{h}^1_{(2)}, \mathbf{h}^2_{(2)}, \mathbf{h}^2_{(2)}, \mathbf{h}^3)] \\ & - \mathbf{E}_{\mathbf{y},\mathbf{h}^1_{(1)},\mathbf{h}^1_{(2)},\mathbf{h}^2_{(1)},\mathbf{h}^2_{(2)},\mathbf{h}^3|\mathbf{x}_{i(1)},\mathbf{x}_{i(2)}}[\frac{\partial}{\partial\theta} E(\mathbf{y}, \mathbf{x}_{i(1)}, \mathbf{x}_{i(2)}, \mathbf{h}^1_{(1)}, \mathbf{h}^1_{(2)}, \mathbf{h}^2_{(2)}, \mathbf{h}^2_{(2)}, \mathbf{h}^3)].\end{aligned} \tag{4.40}$$

The learning procedure is similar to the one explained for class DBM. For the first expression, we mean field (MF) approximation to estimate the gradient. To perform MF, for each observation i, we will find the mean expectation of each layer given $\mathbf{x}_{i(1)}, \mathbf{x}_{i(2)}$, and $\mathbf{y}_i$. Also, to approximate the expectation in second equation, we suggest using Gibbs Sampling method to sample from $\mathbf{y}, \mathbf{h}^1_{(1)}, \mathbf{h}^1_{(2)}, \mathbf{h}^2_{(1)}, \mathbf{h}^{(2)}_{(2)}, \mathbf{h}^3$ observations given $\mathbf{x}_{i(1)}, \mathbf{x}_{i(2)}$. Likewise, for predicting a new observation,we approximate the expectation of the hidden values for each class label, and we use these approximation to calculate the probability of observing each class label. The models in Equation. 4.39 and 4.38 can be easily extended for more than two classes.

4.4 Case Study

In this section, we implement our proposed ClassDBM approach on two benchmark datasets MNIST [70] and NORB [71]. Also, we utilize the multimodal ClassDBM method to predict the quality of audio advertisement multimedia data.

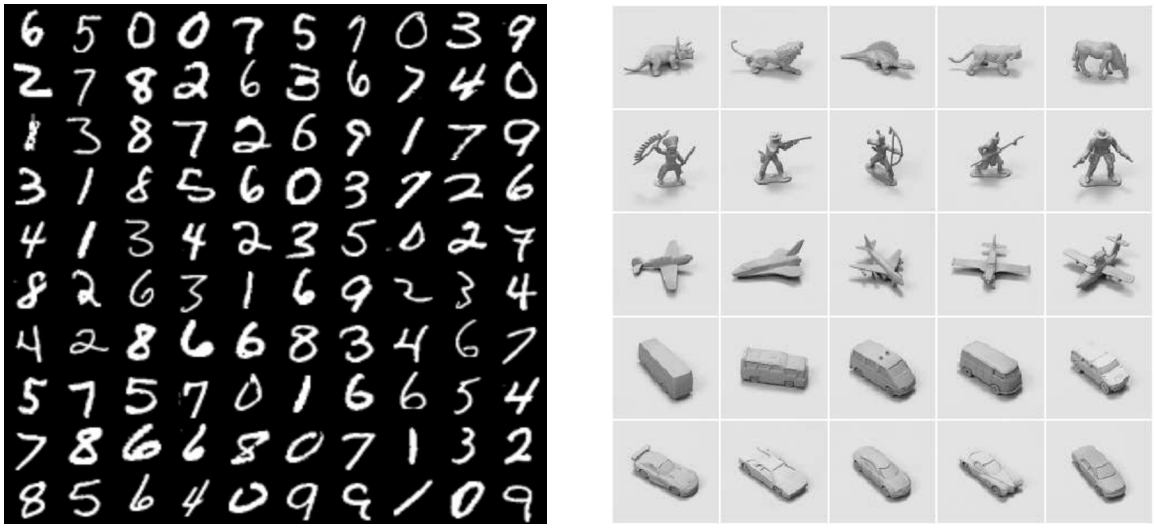
4.4.1 Benchmark Data

In this section, two famous benchmark data are used to evaluate the performance of our methodology. These data has been commonly used in literature for classification of image data. To evaluate the prediction efficancy we will use misclassification error rate which is the percent of misclassified records out of the total records in the test data. We compare our approach with ClassRBM, RBM+NN, and DBM+NN methods. For RBM+NN, we first use a generative RBM to initialize a one layer discriminative feed forward neural network, similar to [9, 68] Similarly, in DBM+NN approach, we first use a generative DBM with two hidden layers and then use the learned weights to initialize a two layer discriminative feed forward neural net.

MNIST

The MNIST (Modified National Institute of Standards and Technology) database is a database including images of handwritten digits and their corresponding labels. This database is used frequently for evaluating various image processing systems. An example of 100 observations from this dataset is presented in figure 4.6 (a). This dataset includes $60,000$ examples for training and $10,000$ for validation. All digit images has a fixed size of 28×28 pixels. The images are greyscale; i.e. each pixel is depicted with a value between 0 and 255, where 0 is black, 255 is white and anything in between is a distinct shade of grey. To learn the model, we split the training data into $50,000$ data for training and $10,000$ data for validation. To implement our proposed approach, we used a two layer ClassDBM with number of hidden units as 512 and 256 for the first and second hidden layer respectively. Also, we set the number of Gibbs sampling step as 3, maximum number of mean field iterations as 50, and mean field tolerance as $1e-7$. We use batch size of 100, and number of epochs of 200. We tune the model parameters using the validation observation, and use the learned model to predict the digit number for test data. To evaluate the prediction accuracy of our model, we use misclassification error rate. We compare the performance of our method with three benchmark methods. ClassRBM method using one hidden layer of 512 units, a

RBM+NN using one hidden layer of 512 units, and DBM+NN using two hidden layers of 512, and 256 units. We use the same number of epochs and number of minibatches. The results are given in Table 4.1. As we can see, our proposed approach provides the lowest error rate, while RBM+NN and DBM+NN are having the highest error rate.



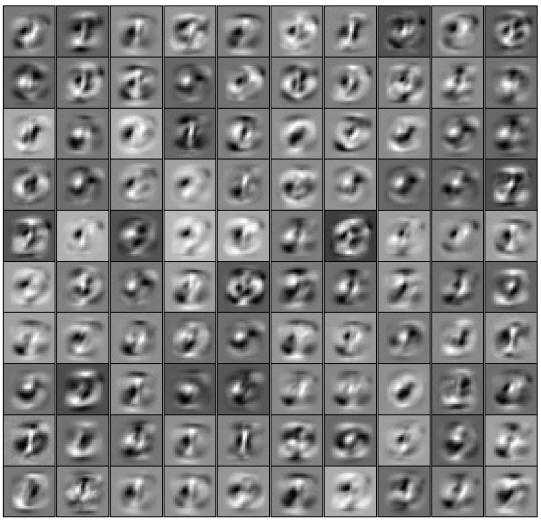
(a) MNIST data (b) NORB data

Figure 4.6: Examples of MNIST and NORB datasets

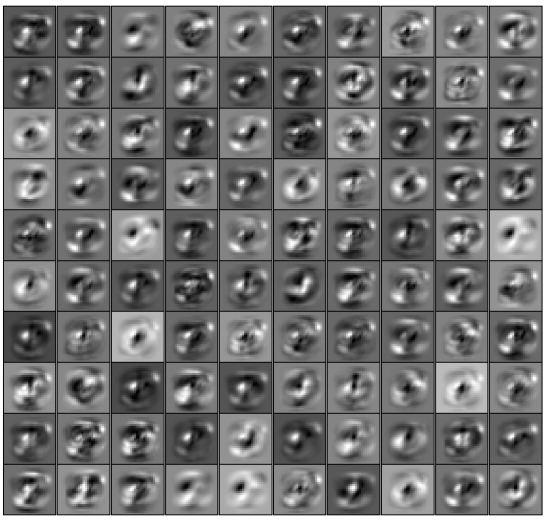
By illustrating the rows of the learned weight matrices $\mathbf{W}_1$, $\mathbf{W}_1\mathbf{W}_2{}^T$, and $\mathbf{W}_1\mathbf{U}^T$, we can apprehend the discriminative strength of the ClassDBMs. Each row can act as a filter feature. These learned features are illustrated in 4.7. As it is shown, these weights can capture particular shapes of digits. More importantly, the weights for discriminative layers are highly illuminating with respect to the particular shape of each digit. In figure 4.7(c), we can clearly observe the shape of digits zero, two, three, and six. The shape of other digits are more noisy because of the overlap the shapes have with each other.

NORB

NORB (NYU Object Recognition Benchmark) dataset contains images taken from fifty different toy objects. Where each ten objects belongs to one of five generic classes including cars, trucks, planes, animals, and humans [71]. Moreover, each image is taken from a different viewpoint and in a specific lighting condition. The dataset is split into training and



(a) First Layer



(b) Second Layer

(c) Discriminative layer

Figure 4.7: First 100 layers of ClassDBM for each layer

testing. The training set contains $24,300$ images. Where images are taken from 25 objects, 5 in each class. The test set contains $24,300$ images of the remaining 25 objects. Each image is accompanied with a label corresponding to its generic class. To tune the model parameters, from the training data, $4,300$ images were randomly selected for validation. All images have 96×96 pixels with greyscale values between 0 and 255. To implement our proposed approach, we used a two layer ClassDBM with number of hidden units as 1024 and 512 for the first and second hidden layer respectively. Also, we set the number of Gibbs sampling step as 3, maximum number of mean field iterations as 50, and mean field tolerance as $1e-7$. We use batch size of 100, and number of epochs of 200. We tune the model parameters using the validation observation, and use the learned model to predict the digit number for test data. To evaluate the prediction accuracy of our model, we use misclassification error rate. We compare the performance of our method with three benchmark methods. ClassRBM method using one hidden layer of 1024 units, a RBM+NN

using one hidden layer of 1024 units, and DBM+NN using two hidden layers of 1024, and 512 units. We use the same number of epochs and number of mini-batches. Results are shown in Table 4.1. As the results shows, the ClassDBM has lowest misclassification error rate in comparison to other methods.

Table 4.1: Classification Error rate for Benchmark Datasets

	MNIST	**NORB**
ClassDBM	**5.81**	**10.39**
ClassRBM	6.32	11.95
DBM+NN	13.41	19.13
RBM+NN	15.21	28.39

4.4.2 Audio Advertisement

Audio advertising is abundantly used by online music streaming services. When users are listening to music, between two different songs, they can be exposed to audio messages from advertisers. This form of ads are called audio ads and they can have a high impact on user listening experience. Given an opportunity space for ads, typically there is an ad retrieval and bidding process. After, the eligible ads are ordered based on the amount of money the advertisers are willing to pay times the quality of the ads. The challenge is how to predict the quality of the audio ads, and what are the key components that create engaging audio ads. In this section, we use a set of audio ads to learn the model that predicts the quality of ads. The dataset includes $7k$ observation. For each observation, advertisement data includes audio signals, ad's accompanying image, and the text script derived from audio signals. Also, each observation has a binary label indicating the quality of the ad as "good" or "bad." The labels are generated based on audio Long Click Rate (LCR). LCR is the ratio of times when an ad was long clicked (i.e., the user clicked on the ad and stayed on the corresponding website for a time higher than a threshold). The ads with higher LCR are indicated as "good", and the ads with lower LCR are indicated as "bad". In the following, we describe each ad's modality, and how we pre-processed each

modality for our multimodal ClassDBM method.

Advertisement Images

When playing an audio advertisement, an image is accompanied with each audio ad. The image contains visual information about the ad that is playing, and users can click on the image to go to the advertisement's website for more information. It's important to include these images when modeling advertisement quality. These images usually vary in length and resolution. To have same input length for all data points, we first resize all images into a 128×128 pixels. Moreover, to implement our binary RBM on image data, we convert the color images into greyscale. Hence, each image pixel can be represented with a value between 0 and 255. Moreover, we vectorize each image as one vector of 16384×1 as the input of our model.

Advertisement Audio files

Raw audio waveforms usually have extremely high temporal resolution. Therefore, when working with audio files, it's a common practice to convert audio waveforms to audio spectrograms, and use the spectrograms as an input of the learning model. Audio spectrograms are $R_{F \times N}$ matrices where F is the number of frequency bins and N is the total time frames. Inspired by works in the music deep learning [91], we first prepare our audio files using F = 100 frequency bin, log-compressed constant-Q transforms (CQT) [101] for all the mp3 ads in our dataset. We used librosa library [84] for this. The parameters to obtain the spectrograms are audio sampling rate at 44100Hz, hop length of 1024 samples, and 12 bins per octave. Furthermore, log-amplitude scaling is applied to the CQT spectrograms with a power law of 2 and scaling of 0.1. Since audio files have different lengths, we need to normalize the sizes of all the spectrograms to train with mini-batches of data points such that the gradients become more stable. To do so, and similar to [**ebrahimi2018predicting**, 91], we sample three 10-seconds patches from each ad, resulting in a fixed size N input to

the network. This procedure is only used in training because during training we need to place input in tensors with fix number of features. During inference, the network can make predictions from full (i.e., not patched and variable N) audio ads.

Text Data

Although audio scripts were not available for our analysis, we can extract the script using available speech to text converter methods. One recent technology is the Google Speech to text API that can convert audio files to text by applying powerful neural network models on them. This API shows a promising accuracy in extracting the script. Also, it is very straightforward to use and also is able to distinguish 120 languages and variants, to support most languages throughout the world. After obtaining the script of each audio file as a text file, pre-processing the text information is the essential next step for obtaining a functional classification model. For that it's common to use pre-trained word embeddings. Among these embeddings, the word2vec embedding dictionary is one of the most frequently used tool for text data. This embedding is trained using 100 million words from Google news [87]. Using the word2vec each word is converted into a 300-dimensional vector. Moreover, similar to other modalities, we need to have inputs with same length for training. However the length of the scripts in our database vary between 1 and 122 words. To obtain same input lengths, we zero-padded all scripts to the longest length as suggested in literature [63, 46]. Each script was therefore converted to a 122×300 matrix.

Multi modal

For the multimodal analysis, we assume three modalities of audio ads as images, audio spectrograms, and text files. Since each modality has its unique characteristics, it is much harder to discover the relationships across different modalities than relationships among features in the same modality. To evaluate the proposed multimodal ClassDBM, we implement it on "Audio+Text", "Audio+Image", "Image+Text", and "Audio+Image+Text".

To evaluate the efficacy of our proposed model, we evaluate unimodal ClassDBM on each modality as well as multimodal ClassDBM on multiple modalities. For unimodal ClassDBMs we use two hidden layers, while for multimodals we use three hidden layers. The number of hidden units in each layer is presented in Table 4.2. For the multimodal architectures, the number of units in the first and second hidden layers for each modality are same as the number of units (in the first and second hidden layers) in their corresponding unimodal architecture. For unimodal methods, to compare the performance of our deep architecture with the shallow architecture, we also implement ClassDBM on Audio Data, Text Data, and Image Data separately. We will use Class DBM with two layers with

Moreover, since Convolutional neural networks (CNN) are common approaches for learning audio and image data, we implement CNN on Audio Data and Image Data as well. For Text observations, a well known deep learning approach is Long Short Term Memory (LSTM) which we implement on our Text Data. Moreover, To evaluate the method, we calculate Area Under the ROC Curve (AUC) on test data. AUC is a popular quality metric when working with binary classifiers which is "equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance" [32]. AUC values ranges between 0.0 and 1.0. While an informative classifier should have an AUC much larger than 0.5, the greater it is the better the classifier is. The results are shown in Table 4.4. As the results suggests, for models with one modality, ClassDBM always outperforms ClassRBM. Also the ClassDBM is working better than CNN on image data. Moreover, the result shows the advertisement text and audio data are the most informative modalities. As the table shows, we gain the highest AUC incorporating these two modalities in a multimodal ClassDBM approach.

4.4.3 Inferences

One drawback of deep learning methods is the lack of straightforward interpretation in such methods. To deal with this problem, one way to gain intuitions about our model is

Table 4.2: Network Architecture For Each Model

	$\mathbf{h}^1$	$\mathbf{h}^2$	$\mathbf{h}^3$
Audio+Text	-	-	512
Audio+Image	-	-	512
Image+text	-	-	512
Audio+Image+Text	-	-	1024
Text Data	1024	512	-
Audio Data	2048	1024	-
Image Data	1024	512	-

to inspect the ads that are correctly predicted as having good versus poor quality. This is done by clustering ads based on the last layer of the ClassDBM network of each of the modelities. The last hidden layer of a neural network can be seen as a learned representation of the data. This representation is learned toward minimizing the loss function where in our case it's minimizing $p(\mathbf{y}|\mathbf{x})$. Hence, it has information toward the objective function we prefer. Therefore, by clustering the ads based on this new representation, we can separate ads toward our objective function. By scrutinizing the ads of clusters that are primarily true-positives or true-negatives, we can draw inferences about the characteristics that makes a "good" or "bad" quality audio ad. To perform this, we cluster the ads based on the last hidden layer of ClassDBM on Audio Data. Similar analysis can be applied on text data.

Audio Analysis

listening to the clusters with highest and lowest quality we can gain insights about differences between these clusters. This qualitative analysis showed that *Good Ads* have clear, mid-paced, solo voice. They contain moderately varied, non-monotone speaker expression with moderate excitement. The speech is conversational between the speaker and the audience (i.e., story telling, call-to-action) or between two isolated speakers in the ad. If there is background music, it is basic. There is a good balance between background and foreground sounds, and there are almost no sound effects. *Poor quality ads* have faster paced language, long winded explanations of products, and very monotone expression. Many of them had

loud backing music and jarring sound effects that sometimes obscure the speech. Finally many of these were also lower quality recordings with distortion and compression effects.

Table 4.3: AUC for predicting audio advertisement quality

	ClassRBM	**CNN**	**LSTM**	**ClassDBM**
Audio+Text	-	-	-	**0.81**
Audio+Image	-	-	-	0.72
Image+text	-	-	-	0.69
Audio+Image+Text	-	-	-	0.64
Text Data	0.71	-	0.73	0.78
Audio Data	0.59	0.79	-	0.71
Image Data	0.52	0.51	-	0.52

Table 4.4: Run time analysis for one epoch (seconds)

	ClassRBM	**CNN**	**LSTM**	**ClassDBM**
Audio+Text	-	-	-	
Audio+Image	-	-	-	
Image+text	-	-	-	
Audio+Image+Text	-	-	-	
Text Data	0.71	-	0.73	0.78
Audio Data	0.59	0.79	-	0.71
Image Data	0.52	0.51	-	0.52

4.5 Conclusion and future study

To conclude, we proposed a new estimation approach which is an extension of ClassRBM method. The new methodologies, named as Classification DBM (ClassDBM) allows the ClassRBM to be applied on deeper architectures. We showed that, the discriminative loss function in ClassRBM methodology become interactable when the number of hidden layers is more than one. Hence, we introduced a new algorithm for estimating the interactable loss function. This algorithm approximate the gradients of loss function using mean field inference and gibbs sampling. Moreover, we showed that after learning the model, predicting the class label of a new observation is not straightforward since the values of hidden layers are unknown. Hence, we presented an approximation method based on mean field

inference to calculate the probability of observing each class label and choose the class label with highest probability as the predicted value. Furthermore, we showed that our approach could be extended to a multimodal framework. Afterward, to show the effectiveness of our methodology, we implemented the proposed approach on two distinguished image datasets of MNIST, and NORB. We showed the superiority of our approach in comparison with ClassRBM and traditional classification RBM and DBM methods. Moreover, we implemented the ClassDBM and multimodal ClassDBM on audio advertisement data to predict the advertisement's quality and achieved highest prediction AUC using the multimodal ClassDBM model.

For future studies, we can utilize hybrid models in the objective function. Hybrid models were used as an extension of ClassRBM [68] when the loss function includes the generative model as well as discriminative model. In such models, the generative loss function is added to the discriminative loss function with a tuning parameter that can be adjusted. The proposed ClassDBM approach can be extended to Hybrid ClassDBM by including the generative model in the loss function. Moreover, in this study we focused only on binary observations; however, to fit the real-world observations more appropriately, it is best to extend the method formulations for other observation distributions such as Normal and Count Data.

CHAPTER 5
CONCLUSION AND FUTURE DIRECTIONS

This dissertation focused on developing models for monitoring and prediction of high dimensional multi-stream and multimedia data. From a large pool of research problems in this area, three main challenging problems were studied, and novel methodologies were developed. New research contributions were made for each methodology. They were experimentally validated, and real-world applications were demonstrated through case studies. In the following, the developed research methodologies and their contributions are summarized. Also, future research directions are suggested.

In Chapter 2, a novel monitoring and diagnosis approach based on PCA is proposed. The proposed methodology seamlessly integrates monitoring and diagnostics for high dimensional multi-stream data. This chapter includes two modules one for monitoring and one for diagnosis. In the monitoring module, we first showed that adaptively selected PCs would be more effective in comparison with the PCA monitoring methods that choose the PCs with the highest variations. Therefore, we proposed a new PCA-based monitoring approach names as Adaptive PC selection (APC) monitoring. Moreover, we showed that the adaptively selected PCs would work in different scenarios of shifts, such as a shift in the correlated variables, a shift in a process with block-diagonal covariance, or a shift in the direction of any arbitrary PC. The shift mentioned above are the common shift scenarios in most high-dimensional processes in which our proposed APC method significantly outperformed the existing state-of-the-art methods. Hence, APC is broadly applicable. Importantly, in any of the scenarios, the conventional approach of monitoring high-PC is shown to have a weaker performance. In the diagnosis module, we first discuss the challenge in finding the shift causing process variable after PCA-based monitoring. The challenge lies in isolating the process variable from the shift signaling PC, which in turn is a combination

of several process variables. We show that Compressed Sensing principles can be used in this premise. Also, we used the CS principle to formulate an adaptive Lasso estimation. This formulation takes the eigenvectors and principal components (after a shift) as inputs and yields the process variables that caused the shift. Our experimental validations showed that PCSR performs markedly better than the state-of-the-art. Furthermore, we show the practical applicability and validity of our methods via real-world case studies. The first case study is on Steel rolling process, in which we find the proposed APC to detect the shift faster than all the other methods. Moreover, the PCSR diagnosis approach detects the change pixels better than the existing method. In another case study, we monitored wine quality and diagnosed the shift to identify the shift causing process variables. Our monitoring approach was again faster, and our diagnosis approach could find an additional shifted process variable,*sugar*, that was missed by the existing state-of-the-art diagnostics approach. In this paper, we have focused on monitoring and diagnosing of the shifts in the mean of the process, while we can extend the methodology to the cases where the shifts occur in the covariance matrix. Moreover, in future, we also aim to extend this approach to Dynamic PCA methods.

In the chapter 3, a new methodology for dynamically monitoring sparse network is proposed. For this, we first showed that common network monitoring methods do not take into account the sparsity in the model. While in many communication networks, not all nodes are connected with each other; hence, there are several zeros (no connectivity) in the edge observations. Therefore, in our methodology, we employ hurdle models that can take into account the excess zeros in the model. Moreover, networks are changing dynamically; thus, a proper monitoring method should take into account the dynamics of the model. Also, edges formations are usually a function of edge and nodes attributes. Therefore, in our proposed methodology, we combine a state space model with the Hurdle model to capture temporal dynamics of the edge formation process, which is modeled as a function of the node and edge attributes and estimated using an extended Kalman Filter. Elements from

statistical process control, such as Exponential Weighted Moving Average control charts, are used to monitor the network sequence in real time in order to distinguish gradual change resulting from the typical edge dynamics from abrupt changes in trading patterns that are caused by fundamental changes in market conditions. We demonstrated the efficacy of our proposed methodology by performing simulation studies. In such, we compared our method with traditional monitoring approaches and showed that our method could detect a change faster in comparison to other methods. More specifically we showed that if the change affects the model sparsity, our proposed methodology significantly outperforms the existing method in the literature. This shows the strength of our method in considering and modeling sparsity in networks. Moreover, for the case study, we focus on modeling the network connections in financial institutions. The interconnectedness of financial institutions can function as a mechanism for the propagation and amplification of shocks throughout the economy, thus contributing to financial crises. As such, network analysis has become a critical tool to assess interconnectedness and systemic risk levels. Employing our method, we find that the proposed methodology would have raised alarms to regulators before several key events and announcements by the European Central Bank during the 2007-2009 financial crisis, proving promise of the approach as an early warning system. For future studies, similar networks can be constructed from other types of markets [11, 1] or even inferred statistically from stock market data [13, 26]; applying the proposed methodology to different markets will require changing the node and edge attributes as well as potentially the Hurdle model itself since the networks may exhibit different or additional properties to sparsity. Further, when the market participant ID is known (in our case it was hidden to preserve confidentiality), it would be useful in practice to monitor several networks (markets) simultaneously since it becomes possible to link activity across markets. Simultaneous monitoring of multiple networks creates several challenges, from visualization of the data and results to detecting changes in multivariate distributions. As such, our work represents the first rigorous application of monitoring techniques beyond tracking network statistics to

financial networks, and the results demonstrate the proposed approach could be a valuable starting point to utilize and extend when monitoring the financial system.

In Chapter 4, a novel deep learning approach for predicting multimedia data labels is proposed. The method is an extension to Classification Restriction Boltzmann Machine (ClassRBM). ClassRBM extends the generative RBM methodologies for application in the discriminative problems. However, due to estimation intractability, a useful deep extension of ClassRBM has not been used in the literature. In this chapter, we introduced a new algorithm for estimating the interactable loss function. This algorithm approximates the gradients of loss function using mean field inference and Gibbs sampling. Also, when the number of hidden layers is higher than one, we showed that predicting the class label of a new observation is not straightforward. Therefore, an approximation prediction methodology is proposed. This method approximates each class label probability using mean field inference and selects the class label with the highest probability as the predicted value. Moreover, we extended the ClassDBM to a multimodal framework where the input is available in different modalities. The efficacy of the ClassDBM is studied by implementing the method on two well-studied datasets. For this, we implemented the method on MNIST and NORB data and showed the superiority of our model in comparison with ClassRBM and traditional classification RBM and DBM methods. Lastly, we implemented the ClassDBM and multimodal ClassDBM on audio advertisement data to predict the advertisement's quality. We compared our method with common deep learning methodologies as well as ClassRBM method. We showed that the highest prediction AUC was achieved using the multimodal ClassDBM model. For future studies, we wish to extend our discriminative class DBM into the hybrid (generative+discriminative) DBM method. Hybrid models were introduced as an extension of ClassRBM [68] when the loss function includes the generative model as well as the discriminative model. In a hybrid RBM model, the generative loss function is added to the discriminative loss function with a tuning parameter that can be adjusted. The proposed ClassDBM approach can be extended to Hybrid Class-

DBM by adding the generative model in the loss function. Furthermore, we are planning to expand the binary ClassDBM to cases where inputs take real values. For this reason, in future studies, we can work on extending the model to Normal ClassDBM.

Appendices

APPENDIX A

SUPPLEMENTARY MATERIAL OF CHAPTER 2: FIRST AND SECOND MOMENTS OF THRESHOLDED STATISTIC

The first moment of $\tilde{d}$ is calculated as,

$$\begin{aligned} E(\tilde{d}) &= E((d_{ij}-\nu)_+) = E((d_{ij}-\nu)|d_{ij}>\nu)P(d_{ij}>\nu) \\ &= E((d_{ij}|d_{ij}>\nu)p(d_{ij}>\nu) - \nu P(d_{ij}>\nu) \\ &= \int_\nu^\infty dP(d) - \nu P(\chi_1^2>\nu) \\ &= \int_\nu^\infty \{x\frac{1}{\sqrt{2}\Gamma(0.5)}x^{-\frac{1}{2}}e^{-\frac{x}{2}}\}dx - \nu P(\chi_1^2>\nu) \\ &= \frac{1}{\Gamma(0.5)}\{\Gamma(0.5,\frac{\nu}{2}) + e^{-\frac{\nu}{2}}\sqrt{2\nu}\} - \nu P(\chi_1^2>\nu) \end{aligned} \tag{A.1}$$

The calculate the second moment of $\tilde{d}$,

$$\begin{aligned} E(\tilde{d^2}) &= E((d_{ij}-\nu)_+^2) = E((d_{ij}-\nu)_+^2)|d_{ij}>\nu)P(d_{ij}>\nu) \\ &= E(d_{ij}^2|d_{ij}>\nu)P(d_{ij}>\nu) - 2\nu E(d_{ij}|d_{ij}>\nu)P(d_{ij}>\nu) + \nu^2 P(d_{ij}>\nu) \overset{\text{Eq.A.1}}{\rightarrow} \\ &= \int_\nu^\infty \{x^2\frac{1}{\sqrt{2}\Gamma(0.5)}x^{-\frac{1}{2}}e^{-\frac{x}{2}}\}dx - 2\nu\{E(\tilde{d}) + \nu P(\chi_1^2>\nu)\} + \nu^2 P(\chi_1^2>\nu) \\ &= \int_\nu^\infty \{x^2\frac{1}{\sqrt{2}\Gamma(0.5)}x^{-\frac{1}{2}}e^{-\frac{x}{2}}\}dx - 2\nu E(\tilde{d}) - \nu^2 P(\chi_1^2>\nu) \\ &= \frac{1}{\Gamma(0.5)}[3\Gamma(0.5,\frac{\nu}{2}) + e^{-\frac{\nu}{2}}\sqrt{2\nu}(3+\nu)] - 2\nu E(\tilde{d}) - \nu^2 P(\chi_1^2>\nu) \end{aligned} \tag{A.2}$$

APPENDIX B

SUPPLEMENTARY MATERIAL OF CHAPTER 2: CONSISTENCY OF THE DIAGNOSIS METHOD

To prove the consistency of our diagnosis model, we use the derivations in [131]. To show the consistency in adaptive lasso Zou used the following conditions,

Condition B.0.1. *noise have independent identical distribution with mean 0 and variance* σ^2

Condition B.0.2. *for observation matrix* $\boldsymbol{X}$*,and number of observations n,* $\frac{1}{n}\boldsymbol{X}^T\boldsymbol{X} \rightarrow \boldsymbol{C}$*. Where* $\boldsymbol{C}$ *is a positive definite matrix.*

Condition B.0.1 is valid for Eq. 2.5. As we showed, after the transformations, the model noise has iid distribution with variance equal to 1. To show the validity of condition B.0.2 we need to show that this consition holds for $\mathbf{A}^*$ instead of $\mathbf{X}$. We use lemma B.0.1 and its proof to show it.

Lemma B.0.1. *For* $\boldsymbol{A}^*$ *given in Eq. 2.6 ,* $\boldsymbol{C} = \frac{1}{p}\boldsymbol{A}^{*T}\boldsymbol{A}^*$ *is a positive definite matrix*

Proof. The proof is a follows:

$$A^* = \Lambda^{\frac{-1}{2}} A$$

$$\mathbf{C} = \frac{1}{p}\mathbf{A}^{*T}\mathbf{A}^* = \frac{1}{p}A^T\Lambda^{\frac{-1}{2}}\Lambda^{\frac{-1}{2}}A \Rightarrow \mathbf{C} = \mathbf{A}^T\Lambda^{-1}\mathbf{A}$$

To show that C is positive definite matrix it suffice to show that $x^T\mathbf{C}x > 0, \forall x \neq 0$

$$\begin{aligned} x^T\mathbf{C}x &= x^T\mathbf{A}^T\Lambda^{-1}\mathbf{A}x \\ &= z^T\Lambda^{-1}z \\ &= \sum_{i=1}^{p}\frac{1}{\lambda_i}z_i^2 \end{aligned}$$

Where λ_i is the pc score i. Since pc scores are positive, hence:

$$x^T\mathbf{C}x > 0, \forall x \neq 0 \qquad \square$$

Since our model holds the above conditions, we can now show the consistency of our model, as follows:

Theorem B.0.2. *Suppose that λ in Eq. 2.6 varies with p. If $\frac{\lambda_p}{\sqrt{p}} \to 0$, and $\lambda_p \to \inf$, then the adaptive lasso estimate must satisfy the following:*

- *Consistency in variable selection:* $lim_p P(S^* = S) = 1$
- *Asymptotic Normality:* $\sqrt{p}(\hat{\mu}_S^* - \hat{\mu}_S \to_d N(\boldsymbol{0}, \sigma')$

  Where $\sigma' = \boldsymbol{C}_{11}^{-1}$

Proof. for proof of Theorem, please refer to [131] $\square$

APPENDIX C

SUPPLEMENTARY MATERIAL OF CHAPTER 3: ESTIMATING TRANSITION MATRICES FOR THE STATE SPACE MODEL

To estimate $\mathbf{F}$ and $\mathbf{Q}$ of the network, we use a series of in-control network snapshots, i.e., $\mathbf{W}_l;\ l = 1, ..., n$. We first estimate the Hurdle regression coefficients for each snapshot, $\beta_{0,l}$ and $\beta_{0,l}$, by fitting a Logistic Regression model on edge occurrence, and a Positive Poisson Regression model on count numbers where edge value is greater than zero. Then, Vector Autoregressive (VAR) model is fitted on the estimated coefficients to estimate $\mathbf{F}_0$ and $\mathbf{Q}_0$ for Bernoulli model and $\mathbf{F}_1$ and $\mathbf{Q}_1$ for Positive Poisson model. The VAR model is common approach for analysis of multivariate time series, which is an extension of the univariate autoregressive model to multivariate time series [104]. We fit the VAR on coefficients for each model using R [93]. After the fitting, $\mathbf{F}$ is set as the estimated coefficient matrix, and $\mathbf{Q}$ is set as the estimated covariance matrix of the noise. Next, we use these values to fit a state-space Hurdle model and compute the coefficient estimates $\beta_{0,l|l}$ and $\beta_{1,l|l}$. After that, another VAR model is fitted on the estimated coefficients and $\mathbf{F}_0$, $\mathbf{F}_1$ and $\mathbf{Q}_0$, $\mathbf{Q}_1$ are updated. We repeat this procedure until convergence. In summary, the estimation process for each model is is as follows,

Step 1 Initialize $\mathbf{F}$ and $\mathbf{Q}$ using static Ordinary Least Squares and a first order vector autoregressive regression model.

Step 2 Use the EKF with given $\mathbf{F}$ and $\mathbf{Q}$ to obtain the new set of covariances.

Step 3 Fit a VAR(1) model on the coefficients and update $\mathbf{F}$ and $\mathbf{Q}$.

Step 4 Repeat Steps 2 and 3 until convergence or until the maximum number of iterations is reached.

APPENDIX D

SUPPLEMENTARY MATERIAL OF CHAPTER 3: RESULTS FROM E-MID DATA: MONITORING STARTING FROM CRISIS 1

Here we report the estimated coefficients for the Country Difference, and Amount and Rate variables after using the first 20 weeks post-Crisis 1 as online training data. We see a markedly similar pattern to what was reported in the main text, that Country Difference, Amount, and Rate all show patterns consistent with the notion that activity in the interbank market dropped markedly, followed by a mild recovery when the crisis ended [15, 36].

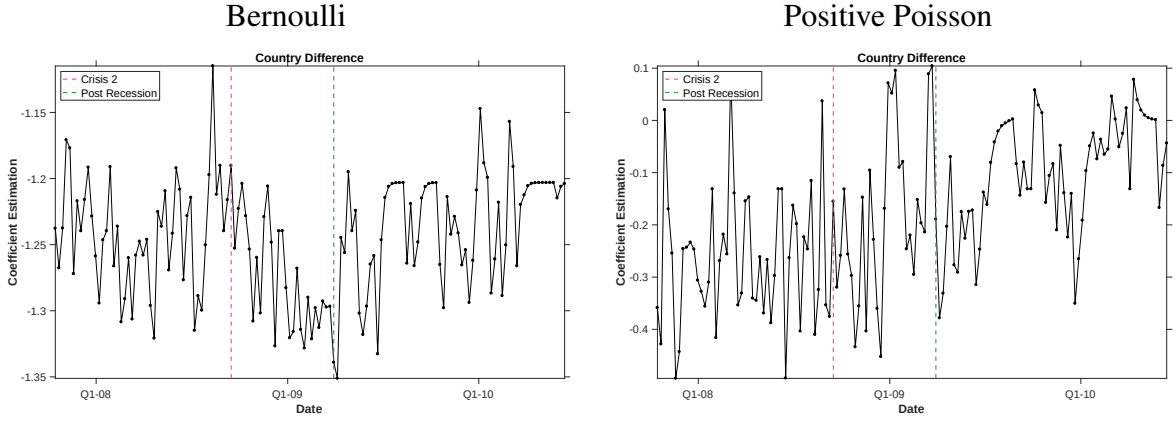


Figure D.1: Estimated Coefficients for Country Difference in the Bernoulli and Positive Poisson models starting from Crisis 1 data.

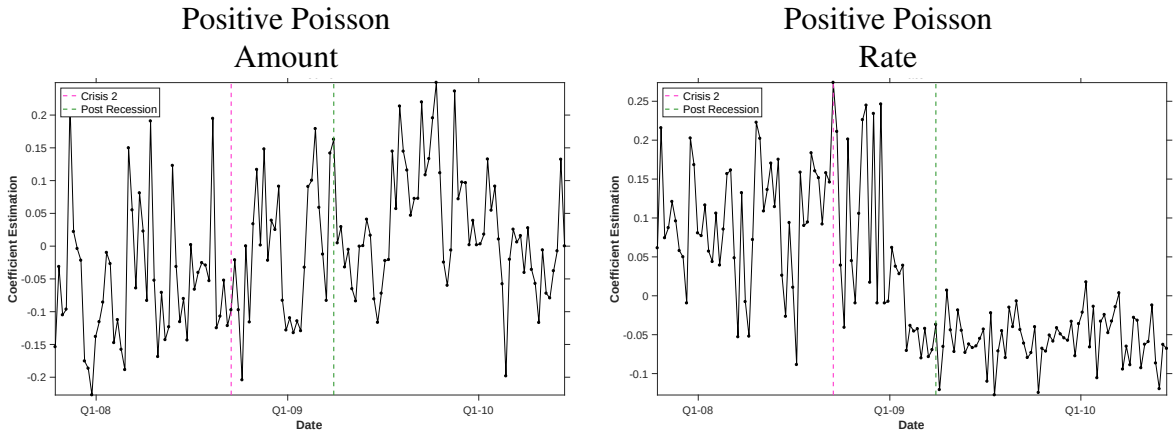


Figure D.2: Estimated Coefficients for Amount and Rate in the Positive Poisson model starting from Crisis 1 data.

REFERENCES

[1] L. Adamic, C. Brunetti, J. H. Harris, and A. Kirilenko, "Trading networks", *The Econometrics Journal*, 2017.

[2] C. C. Aggarwal, "A segment-based framework for modeling and mining data streams", *Knowledge and information systems*, vol. 30, no. 1, pp. 1–29, 2012.

[3] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: A survey", *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[4] C. F. Alcala and S. J. Qin, "Reconstruction-based contribution for process monitoring", *Automatica*, vol. 45, no. 7, pp. 1593–1600, 2009.

[5] M. Antkiewicz, M. Kuta, and J. Kitowski, "Author profiling with classification restricted boltzmann machines", in *International Conference on Artificial Intelligence and Soft Computing*, Springer, 2017, pp. 3–13.

[6] B. Azarnoush, K. Paynabar, J. Bekki, and G. Runger, "Monitoring temporal homogeneity in attributed network streams", *Journal of Quality Technology*, vol. 48, no. 1, p. 28, 2016.

[7] M. Barigozzi and C. T. Brownlees, "Nets: Network estimation for time series", 2016.

[8] S. Basu, S. Das, G. Michailidis, and A. K. Purnanandam, "A system-wide approach to measure connectivity in the financial sector", 2016.

[9] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks", in *Advances in neural information processing systems*, 2007, pp. 153–160.

[10] Y. Bengio and Y. Lecun, "Scaling learning algorithms towards ai", in *Large-scale kernel machines*, MIT Press, 2007.

[11] B. Betancourt, A. Rodríguez, and N. Boyd, "Modelling and prediction of financial trading networks: An application to the nymex natural gas futures market", *arXiv preprint arXiv:1710.01415*, 2017.

[12] C. A. Bhatt and M. S. Kankanhalli, "Multimedia data mining: State of the art and challenges", *Multimedia Tools and Applications*, vol. 51, no. 1, pp. 35–76, 2011.

[13] M. Billio, M. Getmansky, A. W. Lo, and L. Pelizzon, "Econometric measures of connectedness and systemic risk in the finance and insurance sectors", *Journal of Financial Economics*, vol. 104, no. 3, pp. 535–559, 2012.

[14] R. G. Brown and P. Y. Hwang, "Introduction to random signals and applied kalman filtering: With matlab exercises and solutions", *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions, by Brown, Robert Grover.; Hwang, Patrick YC New York: Wiley, c1997.*, 1997.

[15] C. Brunetti, J. H. Harris, S. Mankad, and G. Michailidis, "Interconnectedness in the Interbank Market", *Journal of Financial Economics*, 2018.

[16] M. K. Brunnermeier and L. H. Pedersen, "Market liquidity and funding liquidity", *The review of financial studies*, vol. 22, no. 6, pp. 2201–2238, 2008.

[17] A. C. Cameron and P. K. Trivedi, *Regression analysis of count data*. Cambridge university press, 2013, vol. 53.

[18] E. J. Candes and T. Tao, "Decoding by linear programming", *IEEE transactions on information theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[19] G. Capizzi and G. Masarotto, "A least angle regression control chart for multidimensional data", *Technometrics*, vol. 53, no. 3, pp. 285–296, 2011.

[20] M. A. Carreira-Perpinan and G. E. Hinton, "On contrastive divergence learning.", in *Aistats*, Citeseer, vol. 10, 2005, pp. 33–40.

[21] J. Chang and D. M. Blei, "Hierarchical relational models for document networks", *The Annals of Applied Statistics*, pp. 124–150, 2010.

[22] X. Chen, K. Irie, D. Banks, R. Haslinger, J. Thomas, and M. West, "Scalable bayesian modeling, monitoring and analysis of dynamic network flow data", *Journal of the American Statistical Association*, 2017.

[23] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Modeling wine preferences by data mining from physicochemical properties", *Decision Support Systems*, vol. 47, no. 4, pp. 547–553, 2009.

[24] G. E. Dahl, R. P. Adams, and H. Larochelle, "Training restricted boltzmann machines on word observations", in *Proceedings of the 29th International Conference on Machine Learning*, International Machine Learning Society, 2012.

[25] B. De Ketelaere, M. Hubert, and E. Schmitt, "Overview of pca-based statistical process monitoring methods for time-dependent, high-dimensional data", *Journal of Quality Technology*, vol. 47, pp. 318–335, 2015.

[26] F. X. Diebold and K. Yılmaz, "On the network topology of variance decompositions: Measuring the connectedness of financial firms", *Journal of Econometrics*, vol. 182, no. 1, pp. 119–134, 2014.

[27] R. Dunia and S Joe Qin, "Subspace approach to multidimensional fault identification and reconstruction", *AIChE Journal*, vol. 44, no. 8, pp. 1813–1831, 1998.

[28] European Central Bank, *Ecb financial stability review*, https://www.ecb.europa.eu/press/pr/date/2007/html/pr070615.en.html, Accessed: 2018-01-02, 2007.

[29] ——, *Ecb financial stability review*, https://www.ecb.europa.eu/press/pr/date/2008/html/pr081215.en.html, Accessed: 2018-01-04, 2008.

[30] L. Fahrmeir and H. Kaufmann, "On kalman filtering, posterior mode estimation and fisher scoring in dynamic exponential family regression", *Metrika*, vol. 38, no. 1, pp. 37–60, 1991.

[31] C. Faloutsos, K. S. Mccurley, and A. Tomkins, "Connection subgraphs in social networks", in *SIAM International Conference on Data Mining, Workshop on Link Analysis, Counterterrorism and Security*, 2004.

[32] T. Fawcett, "An introduction to roc analysis", *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[33] Federal Reserve Bank of St. Louis, *Full timeline*, https://www.stlouisfed.org/financial-crisis/full-timeline, Accessed: 2018-01-02, 2018.

[34] S. E. Fienberg, "A brief history of statistical models for network analysis and open challenges", *Journal of Computational and Graphical Statistics*, vol. 21, no. 4, pp. 825–839, 2012.

[35] Financial Crisis Inquiry Commission, *Financial crisis inquiry report*, http://fcic-static.law.stanford.edu/cdn_media/fcic-reports/fcic_final_report_full.pdf, Accessed: 2017-09-01, 2011.

[36] K. Finger, D. Fricke, and T. Lux, "Network analysis of the e-mid overnight money market: The informational value of different aggregation levels for intrinsic dynamic processes", *Computational Management Science*, vol. 10, no. 2-3, pp. 187–211, 2013.

[37] Y. Freund and D. Haussler, "Unsupervised learning of distributions on binary vectors using two layer networks", in *Advances in neural information processing systems*, 1992, pp. 912–919.

[38] D. Fricke and T. Lux, "Core–periphery structure in the overnight money market: Evidence from the e-mid trading platform", *Computational Economics*, vol. 45, no. 3, pp. 359–395, 2015.

[39] ——, "On the distribution of links in the interbank network: Evidence from the e-mid overnight money market", *Empirical Economics*, vol. 49, no. 4, pp. 1463–1495, 2015.

[40] M. R. Gahrooei and K. Paynabar, "Change detection in a dynamic stream of attributed networks", *arXiv preprint arXiv:1711.04441*, 2017.

[41] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, "Big data preprocessing: Methods and prospects", *Big Data Analytics*, vol. 1, no. 1, p. 9, 2016.

[42] P. V. Gehler, A. D. Holub, and M. Welling, "The rate adapting poisson model for information retrieval and object recognition", in *Proceedings of the 23rd international conference on Machine learning*, ACM, 2006, pp. 337–344.

[43] A. Goldenberg, A. X. Zheng, S. E. Fienberg, E. M. Airoldi, *et al.*, "A survey of statistical network models", *Foundations and Trends® in Machine Learning*, vol. 2, no. 2, pp. 129–233, 2010.

[44] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, `http://www.deeplearningbook.org`.

[45] J. T. Grogger and R. T. Carson, "Models for truncated counts", *Journal of applied econometrics*, vol. 6, no. 3, pp. 225–238, 1991.

[46] Y. Gu, X. Li, S. Chen, J. Zhang, and I. Marsic, "Speech intention classification with multimodal deep learning", in *Canadian Conference on Artificial Intelligence*, Springer, 2017, pp. 260–271.

[47] D. J. Hand, "Statistical analysis of network data: Methods and models by eric d. kolaczyk", *International Statistical Review*, vol. 78, no. 1, pp. 135–135, 2010.

[48] R. Hassanzadeh, R. Nayak, and D. Stebila, "Analyzing the effectiveness of graph metrics for anomaly detection in online social networks", in *International Conference on Web Information Systems Engineering*, Springer, 2012, pp. 624–630.

[49] J. Haupt and R. Nowak, "Signal reconstruction from noisy random projections", *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 4036–4048, 2006.

[50] N. A. Heard, D. J. Weston, K. Platanioti, D. J. Hand, *et al.*, "Bayesian anomaly detection methods for social networks", *The Annals of Applied Statistics*, vol. 4, no. 2, pp. 645–662, 2010.

[51] G. Hinton, "A practical guide to training restricted boltzmann machines", *Momentum*, vol. 9, p. 1, 2010.

[52] G. E. Hinton, "To recognize shapes, first learn to generate images", *Progress in brain research*, vol. 165, pp. 535–547, 2007.

[53] ——, "Training products of experts by minimizing contrastive divergence", *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[54] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks", *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[55] ——, "Replicated softmax: An undirected topic model", in *Advances in neural information processing systems*, 2009, pp. 1607–1614.

[56] A. Hyvarinen, "Connections between score matching, contrastive divergence, and pseudolikelihood for continuous-valued variables", *IEEE Transactions on neural networks*, vol. 18, no. 5, pp. 1529–1531, 2007.

[57] J. E. Jackson and G. S. Mudholkar, "Control procedures for residuals associated with principal component analysis", *Technometrics*, vol. 21, no. 3, pp. 341–349, 1979.

[58] N. Jaitly and G. Hinton, "Learning a better representation of speech soundwaves using restricted boltzmann machines", in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, IEEE, 2011, pp. 5884–5887.

[59] P. Ji, J. Jin, *et al.*, "Coauthorship and citation networks for statisticians", *The Annals of Applied Statistics*, vol. 10, no. 4, pp. 1779–1812, 2016.

[60] S Joe Qin, "Statistical process monitoring: Basics and beyond", *Journal of chemometrics*, vol. 17, no. 8-9, pp. 480–502, 2003.

[61] R. E. Kalman *et al.*, "A new approach to linear filtering and prediction problems", 1960.

[62] L. L. C. Kasun, H. Zhou, G.-b. Huang, and C. M. Vong, "Representational learning with elms for big data", 2013.

[63] Y. Kim, "Convolutional neural networks for sentence classification", *arXiv preprint arXiv:1408.5882*, 2014.

[64] A. Krishnamurthy, "Amplification mechanisms in liquidity crises", *American Economic Journal: Macroeconomics*, vol. 2, no. 3, pp. 1–30, 2010.

[65] P. N. Krivitsky and M. S. Handcock, "A separable model for dynamic networks", *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 76, no. 1, pp. 29–46, 2014.

[66] A. Krizhevsky, G. Hinton, *et al.*, "Factored 3-way restricted boltzmann machines for modeling natural images", in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 621–628.

[67] D. Lambert, "Zero-inflated poisson regression, with an application to defects in manufacturing", *Technometrics*, vol. 34, no. 1, pp. 1–14, 1992.

[68] H. Larochelle and Y. Bengio, "Classification using discriminative restricted boltzmann machines", in *Proceedings of the 25th international conference on Machine learning*, ACM, 2008, pp. 536–543.

[69] N. Le Roux and Y. Bengio, "Representational power of restricted boltzmann machines and deep belief networks", *Neural computation*, vol. 20, no. 6, pp. 1631–1649, 2008.

[70] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[71] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting", in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, IEEE, vol. 2, pp. II–104.

[72] T. S. Lee, D. Mumford, R. Romero, and V. A. Lamme, "The role of the primary visual cortex in higher level vision", *Vision research*, vol. 38, no. 15-16, pp. 2429–2454, 1998.

[73] G. Li, S. J. Qin, and D. Zhou, "A new method of dynamic latent-variable modeling for process monitoring", *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6438–6445, 2014.

[74] Q. Li, J. Zhang, Y. Wang, and K. Kang, "Credit risk classification using discriminative restricted boltzmann machines", in *Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on*, IEEE, 2014, pp. 1697–1700.

[75] W. Li, H. H. Yue, S. Valle-Cervantes, and S. J. Qin, "Recursive pca for adaptive process monitoring", *Journal of process control*, vol. 10, no. 5, pp. 471–486, 2000.

[76] Z.-H. Ling, L. Deng, and D. Yu, "Modeling spectral envelopes using restricted boltzmann machines for statistical parametric speech synthesis", in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, IEEE, 2013, pp. 7825–7829.

[77] K. Liu, Y. Mei, and J. Shi, "An adaptive sampling strategy for online high-dimensional process monitoring", *Technometrics*, vol. 57, no. 3, pp. 305–319, 2015.

[78] L. Ljung, "Asymptotic behavior of the extended kalman filter as a parameter estimator for linear systems", *IEEE Transactions on Automatic Control*, vol. 24, no. 1, pp. 36–50, 1979.

[79] P. M. Long and R. Servedio, "Restricted boltzmann machines are hard to approximately evaluate or simulate", in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 703–710.

[80] J. F. MacGregor, C. Jaeckle, C. Kiparissides, and M Koutoudi, "Process monitoring and diagnosis by multiblock pls methods", *AIChE Journal*, vol. 40, no. 5, pp. 826–838, 1994.

[81] V Mayer-Schnberger, *Big data: A revolution that will transform how we live, work and think, ed. viktormayer-schonberger and kennethcukier*, 2013.

[82] I. McCulloh, "Detecting changes in a dynamic social network", PhD thesis, Carnegie Mellon University, 2009.

[83] I. McCulloh and K. M. Carley, "Detecting change in longitudinal social networks", Military Academy West Point NY Network Science Center (NSC), Tech. Rep., 2011.

[84] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in python", 2015.

[85] Y. Mei, "Efficient scalable schemes for monitoring a large number of data streams", *Biometrika*, vol. 97, no. 2, pp. 419–433, 2010.

[86] ——, "Quickest detection in censoring sensor networks", in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, IEEE, 2011, pp. 2148–2152.

[87] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space", *arXiv preprint arXiv:1301.3781*, 2013.

[88] J. Mullahy, "Specification and testing of some modified count data models", *Journal of econometrics*, vol. 33, no. 3, pp. 341–365, 1986.

[89] J. A. Nelder and R. J. Baker, *Generalized linear models*. Wiley Online Library, 1972.

[90] K. Nishina, "A comparison of control charts from the viewpoint of change-point estimation", *Quality and reliability engineering international*, vol. 8, no. 6, pp. 537–541, 1992.

[91] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation", in *Advances in neural information processing systems*, 2013, pp. 2643–2651.

[92] X. Peng, X. Gao, and X. Li, "An infinite classification rbm model for radar hrrp recognition", in *Neural Networks (IJCNN), 2017 International Joint Conference on*, IEEE, 2017, pp. 1442–1448.

[93] B. Pfaff *et al.*, "Var, svar and svec models: Implementation within r package vars",

[94] A. A. Qahtan, B. Alharbi, S. Wang, and X. Zhang, "A pca-based change detection framework for multidimensional data streams: Change detection in multidimensional data streams", in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2015, pp. 935–944.

[95] S. J. Qin, S. Valle, and M. J. Piovoso, "On unifying multiblock analysis with application to decentralized process monitoring", *Journal of chemometrics*, vol. 15, no. 9, pp. 715–742, 2001.

[96] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak, and F. Herrera, "A survey on data preprocessing for data stream mining: Current status and future directions", *Neurocomputing*, vol. 239, pp. 39–57, 2017.

[97] S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. F. Samatova, "Anomaly detection in dynamic networks: A survey", *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 7, no. 3, pp. 223–247, 2015.

[98] R. Salakhutdinov and H. Larochelle, "Efficient learning of deep boltzmann machines", in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 693–700.

[99] S. Santiago, *Should investors over age 50 own stocks?*, `http://www.bankrate.com/finance/financial-literacy/do-stocks-make-sense-for-the-50-plus-crowd-1.aspx`, Accessed: 2017-09-01, 2009.

[100] L. K. Saul and M. I. Jordan, "Exploiting tractable substructures in intractable networks", in *Advances in neural information processing systems*, 1996, pp. 486–492.

[101] C. Schörkhuber and A. Klapuri, "Constant-q transform toolbox for music processing", 2010.

[102] G. Schwarz *et al.*, "Estimating the dimension of a model", *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[103] D. K. Sewell and Y. Chen, "Latent space models for dynamic networks", *Journal of the American Statistical Association*, vol. 110, no. 512, pp. 1646–1657, 2015.

[104] C. A. Sims, "Macroeconomics and reality", *Econometrica: Journal of the Econometric Society*, pp. 1–48, 1980.

[105] T. A. Snijders, J. Koskinen, and M. Schweinberger, "Maximum likelihood estimation for social network dynamics", *The Annals of Applied Statistics*, vol. 4, no. 2, p. 567, 2010.

[106] R. S. Sparks, "Quality control with multivariate data", *Australian & New Zealand Journal of Statistics*, vol. 34, no. 3, pp. 375–390, 1992.

[107] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep boltzmann machines", in *Advances in neural information processing systems*, 2012, pp. 2222–2230.

[108] J. H. Sullivan, Z. G. Stoumbos, R. L. Mason, and J. C. Young, "Step-down analysis for changes in the covariance matrix and other parameters", *Journal of Quality Technology*, vol. 39, no. 1, p. 66, 2007.

[109] M. Tamura and S. Tsujita, "A study on the number of principal components and sensitivity of fault detection using pca", *Computers & Chemical Engineering*, vol. 31, no. 9, pp. 1035–1046, 2007.

[110] A. G. Tartakovsky, B. L. Rozovskii, R. B. Blažek, and H. Kim, "Detection of intrusions in information systems by sequential change-point methods", *Statistical methodology*, vol. 3, no. 3, pp. 252–293, 2006.

[111] T. Tieleman, "Training restricted boltzmann machines using approximations to the likelihood gradient", in *Proceedings of the 25th international conference on Machine learning*, ACM, 2008, pp. 1064–1071.

[112] J. M. Tomczak, "Application of classification restricted boltzmann machine to medical domains", 2014.

[113] ——, "Prediction of breast cancer recurrence using classification restricted boltzmann machine with dropping", *arXiv preprint arXiv:1308.6324*, 2013.

[114] U.S. Congress, *Public law 110â343*, https://www.gpo.gov/fdsys/pkg/STATUTE-122/pdf/STATUTE-122-Pg3765.pdf, Accessed: 2018-01-02, 2008.

[115] U.S. Department of the Treasury, *Citizens' report*, https://www.treasury.gov/initiatives/financial-stability/reports/Documents/FY2016%20OFS%20CitizensReport_FINAL%20-%2011.22.16.pdf, Accessed: 2017-09-01, 2016.

[116] U.S. Government Accountability Office, *Gao-13-180, financial regulatory reform: Financial crisis losses and potential impacts of the dodd-frank act*, http://www.gao.gov/assets/660/651322.pdf, Accessed: 2017-09-01, 2013.

[117] K. Wang and W. Jiang, "High-dimensional process monitoring and fault isolation via variable selection", *Journal of Quality Technology*, vol. 41, no. 3, p. 247, 2009.

[118] Y. Wang and Y. Mei, "Montoring multiple data streams via shrinkage post-change estimation", *Annals of Statistics*, 2013.

[119] J. A. Westerhuis, S. P. Gurden, and A. K. Smilde, "Generalized contribution plots in multivariate statistical process monitoring", *Chemometrics and intelligent laboratory systems*, vol. 51, no. 1, pp. 95–114, 2000.

[120] S. J. Wierda, "Multivariate statistical process control-recent results and directions for future research", *Statistica Neerlandica*, vol. 48, no. 2, pp. 147–168, 1994.

[121] B. M. Wise, N. Ricker, D. Veltkamp, and B. R. Kowalski, "A theoretical basis for the use of principal component models for monitoring multivariate processes", *Process control and quality*, vol. 1, no. 1, pp. 41–51, 1990.

[122] H. Yan, K. Paynabar, and J. Shi, "Anomaly detection in images with smooth background via smooth-sparse decomposition", *Technometrics*, vol. 59, no. 1, pp. 102–114, 2017.

[123] Y. Yang, "Can the strengths of aic and bic be shared? a conflict between model indentification and regression estimation", *Biometrika*, vol. 92, no. 4, pp. 937–950, 2005.

[124] L. Younes, "On the convergence of markovian stochastic algorithms with rapidly decreasing ergodicity rates", *Stochastics: An International Journal of Probability and Stochastic Processes*, vol. 65, no. 3-4, pp. 177–228, 1999.

[125] L. Yuan and T. Xiao-Chu, "Improved performance of fault detection based on selection of the optimal number of principal components", *Acta Automatica Sinica*, vol. 35, no. 12, pp. 1550–1557, 2009.

[126] H. H. Yue and S. J. Qin, "Reconstruction-based fault identification using a combined index", *Industrial & engineering chemistry research*, vol. 40, no. 20, pp. 4403–4414, 2001.

[127] J. Zhang and J. Cao, "Finding common modules in a time-varying network with application to the drosophila melanogaster gene regulation network", *Journal of the American Statistical Association*, vol. 112, no. 519, pp. 994–1008, 2017.

[128] C. Zou, W. Jiang, and F. Tsung, "A lasso-based diagnostic framework for multivariate statistical process control", *Technometrics*, vol. 53, no. 3, pp. 297–309, 2011.

[129] C. Zou and P. Qiu, "Multivariate statistical process control using lasso", *Journal of the American Statistical Association*, vol. 104, no. 488, pp. 1586–1596, 2009.

[130] C. Zou, Z. Wang, X. Zi, and W. Jiang, "An efficient online monitoring method for high-dimensional data streams", *Technometrics*, vol. 57, no. 3, pp. 374–387, 2015.

[131] H. Zou, "The adaptive lasso and its oracle properties", *Journal of the American statistical association*, vol. 101, no. 476, pp. 1418–1429, 2006.